

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра технічної кібернетики

«На правах рукопису»
УДК 004.043

До захисту допущено:

Завідувач кафедри

_____ Ігор ПАРХОМЕЙ

«__» _____ 2020 р.

Магістерська дисертація

на здобуття ступеня магістра

**за освітньо-професійною програмою «Інформаційне забезпечення
робототехнічних систем»**

зі спеціальності 126 «Інформаційні системи та технології»

**на тему: «Впровадження фізичної симуляції робототехнічних модулів
методом цифрового двійника на основі реальних даних»**

Виконав:

студент II курсу, групи ІК-91мп

Сізов Данило Сергійович _____

Керівник:

доцент, к.т.н., доц.,

Поліщук Михайло Миколайович _____

Консультант з нормоконтролю:

доцент, к.т.н., доц.,

Пасько Віктор Петрович _____

Рецензент:

Головний інженер ТОВ «Хладар-Техсоюз»

Рибаков Дмитро Русланович _____

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних
посилань.

Студент _____

Київ – 2020 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра технічної кібернетики

Рівень вищої освіти – другий (магістерський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційне забезпечення
робототехнічних систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Ігор ПАРХОМЕЙ

«__» _____ 2020 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Сізову Данилу Сергійовичу

1. Тема дисертації «Впровадження фізичної симуляції робототехнічних модулів методом цифрового двійника на основі реальних даних», науковий керівник дисертації доцент, к.т.н., доц., Поліщук Михайло Миколайович, затверджені наказом по університету від « 26 » жовтня 2020р. № 3132-с
2. Термін подання студентом дисертації 19.12.2020
3. Об'єкт дослідження – методи впровадження інфраструктури симуляції методом цифрового двійника у виробництво
4. Вихідні дані – програмне забезпечення для впровадження фізичної симуляції методом цифрового двійника
5. Перелік завдань, які потрібно розробити – аналіз проблеми та існуючих рішень; аналіз і реалізація алгоритму симуляції; розробка програмного забезпечення; дослідження ефективності розробленого програмного забезпечення.
6. Орієнтовний перелік графічного (ілюстративного) матеріалу – шість плакатів
7. Орієнтовний перелік публікацій – нуль публікацій

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Перевірка на співпадіння	доцент Лісовиченко О.І.		
Нормоконтроль	доцент Пасько В.П.		

9. Дата видачі завдання

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Розробка архітектури програмного забезпечення	01.09.2020 - 01.10.2020	
2	Впровадження рішення на виробництві	01.10.2020 - 01.11.2020	
3	Оформлення результатів дослідження	01.11.2020 - 30.11.2020	
4	Нормоконтроль	30.11.2020 - 06.12.2020	
5	Перевірка на співпадіння	01.12.2020 - 10.12.2020	
6	Попередній захист	14.12.2020	
7	Захист	21.12.2020-23.12.2020	

Студент

Данило СІЗОВ

Науковий керівник

Михайло ПОЛІЩУК

АНОТАЦІЯ

У роботі розглянуто проблему в області важкого та середнього виробництва із отриманням актуальної виробничої статистики у реальному часі, показано основні особливості існуючих рішень впровадження інфраструктури для вирішення проблеми статистики, їх переваги та недоліки.

Розроблено систему впровадження у виробництво симуляції методом цифрового двійника, що надає можливості збору даних, їх аналізу та прийняття рішень на основі проведеного аналізу. Дана система може бути використана на виробництві будь-якого масштабу. Система дозволяє підвищити відказостійкість виробництва та підвищити його ефективність за рахунок обробки статистики у реальному часі.

Головною особливістю системи є її гнучкість та адаптивність, а також архітектура, що дозволяє конструювати підсистеми на основі впровадження модулів.

Ключові слова: сервер, цифровий двійник, симуляція, датчик.

Розмір пояснювальної записки – 65 аркушів, містить 11 ілюстрацій, 21 таблиць, 6 додатків.

АНОТАЦІЯ

The paper considers the problem in the field of heavy and medium production with the receipt of current production statistics in real time, shows the main features of existing infrastructure solutions to solve the problem of statistics, their advantages and disadvantages.

A system of implementation of simulation in the production of digital double method has been developed, which provides opportunities for data collection, analysis and decision-making based on the analysis. This system can be used in production of any scale. The system allows to increase the resilience of production and increase its efficiency by processing statistics in real time.

The main feature of the system is its flexibility and adaptability, and such an architecture that allows you to design subsystems based on the implementation of modules.

Keywords: server, digital twin, simulation, sensor.

The size of the explanatory note is 65 sheets, contains 11 illustrations, 21 tables, 6 appendices.

Пояснювальна записка до магістерської дисертації

на тему: Впровадження фізичної симуляції робототехнічних модулів
методом цифрового двійника на основі реальних даних

Київ – 2020 року

Зміст

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	9
ВСТУП	10
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ	11
1.1 Об'єкт та предмет дослідження	11
1.2 Огляд існуючих рішень	17
1.3 Висновки по розділу	18
РОЗДІЛ 2. АНАЛІЗ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ	19
2.1 Обрана база даних	19
2.2 Архитектура базы данных MongoDB NoSQL	19
2.3 Приклад схеми документу company у базі даних	21
2.4 Огляд технологій серверної частини додатку	22
2.5 Огляд технологій користувацького інтерфейсу	24
2.6 Огляд технології для управління розгортанням	25
2.7 Висновки до розділу	27
РОЗДІЛ 3. РОЗРОБКА АЛГОРИТМІЧНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	28
3.1 Алгоритм визначення структури системи симуляції цифрового двійника	28
3.2 Серверний додаток цифрового двійника	32
3.3 Фізична частина процесу інтеграції	36
3.4 Локальний препроцесінг даних	38
3.5 Інтеграція робототехнічних модулів із симуляцією	40
3.5 Висновки по розділу	44
РОЗДІЛ 4. МАРКЕТИНГОВИЙ АНАЛІЗ СТАРТАП-ПРОЕКТУ	46
4.1 Опис ідеї проекту	46
4.2 Технологічний аудит ідеї проекту	48
4.3 SWOT-аналіз проекту	48
4.4. Розроблення ринкової стратегії проекту	54
4.5. Обґрунтування ресурсів та витрат проекту	56
4.6 План робіт та партнери проекту	58
4.7 Грошовий потік та економічна оцінка	59
4.8 Висновки до розділу	61
ВИСНОВКИ	62

Перелік посилань**64**

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ВС – вимірювальна система

ПЛІС – програмована логічна інтегральна схема САК – система автоматичного керування

ЦФ – цифровий фільтр

ФК – фільтр Калмана

ФФ – фільтр Фур'є

АЦП – аналого-цифровий перетворювач

ЦАП – цифро-аналоговий перетворювач

ЦОМ – цифрова обчислювальна машина

ЕОМ – електронна обчислювальна машина

СКП – середньоквадратична похибка

СКВ – середньоквадратичне відхилення

ПУ – початкові умови

КС – кіберфізичні системи

БД – база даних

ВСТУП

Сьогоднішні тенденції спонукають виробництва все більше автоматизувати свої процеси та виробничі ланцюги, що є неодмінним шляхом для повної діджиталізації багатьох сфер сучасного життя.

Але не слід забувати, що діджиталізація – це довгий процес, який складається з багатьох етапів, одним з яких є переведення статистики та даних виробництва із паперу на цифрові носії та автоматична їх обробка.

Звісно, фінальна ціль діджиталізації є автоматизація управління виробництвом або навіть автоматизація самого виробництва, але для цього, потрібно мати можливість оперувати якимись даними, обробляти їх і зберігати, адже рішення повинні прийматись на основі багатьох показників.

Існує велика кількість різноманітних підходів у інтегровані ти діджиталізації підприємства, які містять у собі способи збирати та доставляти дані до основного серверу.

Найуспішніший підхід – збір даних за допомогою примітивних датчиків, що під'єднані до єдиної із головним комп'ютером мережі, адже такий спосіб гарантує масштабованість та низькі ризики експлуатації, саме тому ми будемо використовувати цей спосіб збору.

Але крім отримання даних, система повинна правильно їх обробляти, виконувати “препроцесінг” даних. Найінноваційнішим методом обробки на даний момент є метод симуляції цифрового двійника.

Найбільші виробники все частіше використовують цей метод для обробки даних із своїх виробництв, адже симуляція цифрового двійника є найбільш приближеною до реального світу та до реальних показників методом обробки даних.

Більш того, симуляція методом цифрового двійника дозволяє продовжити життєвий цикл продукту та дає можливість отримувати дані про продукт навіть після продажу продукту

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ

1.1 Об'єкт та предмет дослідження

Четверта промислова революція (або Індустрія 4.0) – це постійна автоматизація традиційних виробничих і промислових практик з використанням сучасних інтелектуальних технологій. Широкомасштабна межмашинного зв'язок (M2M) та Інтернет речей (IoT) інтегровані для підвищення автоматизації, поліпшення зв'язку і самоконтролю, а також для виробництва інтелектуальних машин, які можуть аналізувати і діагностувати проблеми без необхідності втручання людини.

Деякі з основних аспектів, пов'язаних з Індустрією 4.0:

- масова настройка;
- гнучке виробництво;
- відстеження і самосвідомість деталей і продуктів і їх зв'язок з машинами і іншими продуктами;

Таким чином, в контексті Індустрії 4.0 польові пристрої, машини, заводи і окремі продукти будуть підключені до мережі. Кіберфізичні системи (КС) були запропоновані в якості ключової концепції архітектур Індустрії 4.0. КС можна описати як набір фізичних пристроїв, об'єктів і обладнання, які взаємодіють з віртуальним кіберпростіром через мережу зв'язку. Кожна фізична пристрій буде мати свою кібернетичну частину у вигляді цифрового уявлення реального пристрою, кульмінацією якого стане «Цифровий двійник». Таким чином, Digital Twincan відстежує і контролює фізичний об'єкт, в той час як фізичний об'єкт може відправляти дані для поновлення своєї віртуальної моделі.

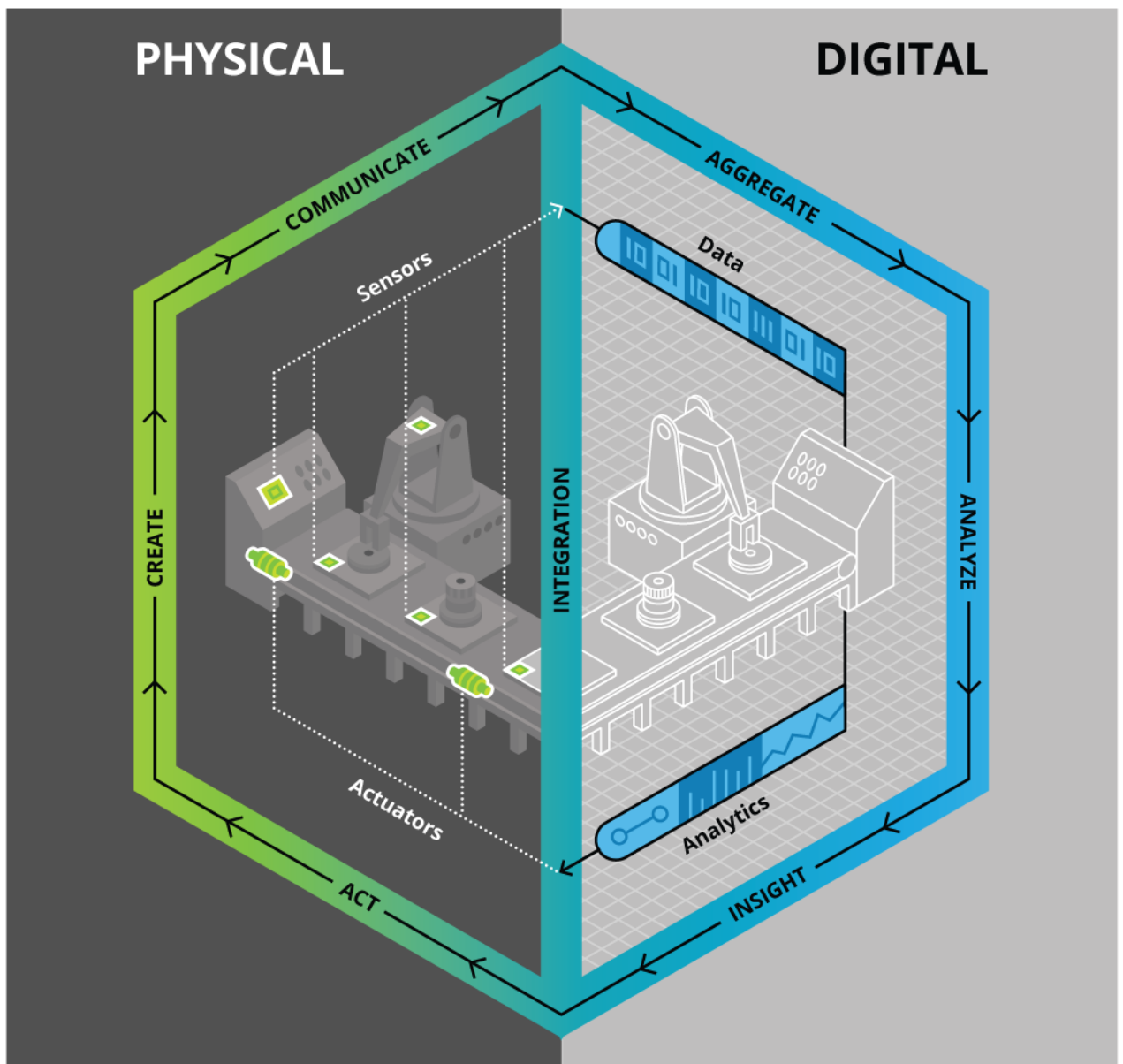


Рисунок 1.1 Стандартна архітектура цифрового двійника

Цифровий двійник є віртуальним поданням реального продукту. Він містить інформацію про продукт з моменту початку його експлуатації до моменту утилізації продукту. Digital Twin можна розглядати як кібер представлення в контексті кіберфізичних систем.

Модель цифрового двійника може складатися з різних моделей і даних. Проблема в тому, як отримати модель, яка об'єднує всі моделі. Модель, яка дозволяє різним системам мати доступ до даних Цифрового двійника. Основна мета розробника у процесі створення універсальної системи цифрового двійника – представити модель, яка пропонує послуги для різних додатків. Для прикладу можна розглянути можливість

використання AutomationML для моделювання симуляції методом цифрового двійника на високому рівні та визначити мету прикладу як показати модель, яка використовує AutomationML для обміну даними між цифровим двійником і іншими системами, а також методологію обміну даними.

По-перше, відомий метод моделювання і зберігання даних – об'єктно-орієнтована парадигма. Однак обмін даними здійснюється за допомогою індивідуальних з'єднань точка-точка.

Інша концепція обміну даними – використання онтологій в якості загальної семантичної основи для моделювання даних.

Друга концепція моделювання – це XML (Extensible Markup Language). Це мова розмітки, який визначає набір правил для кодування документів в форматі, який зручний для читання людиною і комп'ютером. XML може описувати різні типи даних. І він був введений і використовується в різних інструментах і моделях даних

Стандартный формат данных для информации PPRR (продукт, процесс и ресурс) с использованием XML определен и разработан интегратором PLM (управление жизненным циклом продукта), который поддерживает обмен информацией PPR между коммерческими неоднородными системами PLM и другими системами.

Інший метод – Стандарт для обміну даних моделі продукту (STEP). Він регулюється ISO 10303 і являє собою сімейство стандартів, що визначають надійну і перевірену часом методологію опису даних продукту протягом усього життєвого циклу продукту. STEP широко використовується в системах автоматизованого проектування (CAD) і управління даними / життєвим циклом продукту (PDM / PLM). Але STEP в основному зберігає інформацію про продукт, але не дані, необхідні для виробничого моніторингу та управління.

Новий метод – це Computer Aided Engineering Exchange (CAEX). Це абстрактний об'єктно-орієнтований формат даних на основі XML, який

описує реальні або логічні об'єкти підприємства в формі об'єктів даних. Ця концепція корисна не тільки для обміну інформацією з вимірювального пристрою на рівень підприємства, але вже і для обміну інформацією різних додатків на більш високих рівнях, наприклад моніторинг, контроль і аналіз виробництва.

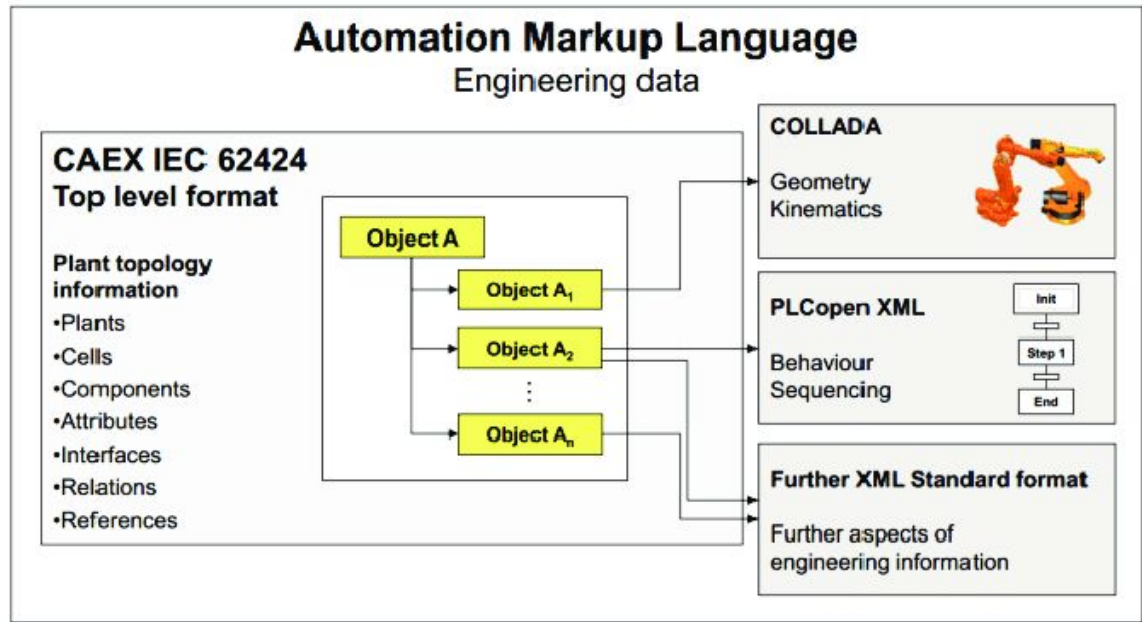


Рисунок 1.2 Приклад схеми мови AutomationML

AutomationML визначає CAEX як метамодель для зберігання інженерних моделей і обміну ними. Він використовувався для моделювання інженерних промислових підприємств і для обміну даними в технологічному ланцюжку виробничих систем. Але ми будемо використовувати цей вид моделі як джерело даних.

Цифровий Двійник – термін «Digital Twin» вперше був представлений широкій публіці в 2010.

У ньому йдеться:

«Цифровий двійник – це інтегроване мультифізическое багатомасштабне моделювання механізму, продукту або системи, в якому використовуються найкращі доступні фізичні моделі, поновлення датчиків, історія операцій, тощо.»

З появою кіберфізичних систем, цифровий двійник з'являється як віртуальне уявлення фізичного продукту, цифрова копія, яка містить всю інформацію і знання про нього. Він якимось чином пов'язаний з фізичною частиною для забезпечення передачі даних з фізичної частини на кібернетичну. Оскільки CPS можна описати як набір фізичних пристроїв, об'єктів і обладнання, які взаємодіють з віртуальним киберпростіром через мережу зв'язку, кібермодель кожного фізичного об'єкта може розглядатися як цифрове представлення реального об'єкта, званого «цифровим двійником».

Професійне об'єднання визначає цифрового двійника як розподілений і децентралізований підхід до управління інформацією про продукт на рівні елемента продукту протягом його життєвого циклу. Таким чином, Digital Twin безпосередньо пов'язаний з концепцією систем управління життєвим циклом продукту (PLM).

PLM об'єднує різні бізнес-операції, які створюють, змінюють і використовують дані для підтримки всіх етапів життєвого циклу продукту: від «початку життєвого циклу» (проектування, виробництво) до «середини життєвого циклу» (використання, обслуговування) і «кінця життя» (переробка, утилізація).

Цифровий двійник включав як статичну, так і динамічну інформацію:

- Статична інформація може бути: геометричні розміри, відомість матеріалів, процеси, тощо.
- Динамічна інформація – це та інформація, яка змінюється з часом протягом життєвого циклу продукту. Цифровий двійник може володіти інтелектом, який обробляє динамічні дані і робить висновки на основі вірогідностей.

Наприклад, інтелектуальний продукт може витягувати інформацію про себе і здатний брати участь або приймати рішення про своє власне майбутнє.

Нижче я привів основні характеристики стандартного інтелектуального продукту:

- вимагає глобальної унікальної ідентифікації;
- здатний до спілкування з навколишнім середовищем;
- може витягувати і зберігати дані про себе;
- здатний брати участь в ухваленні рішень, що мають відношення до його власної долі.

Для Цифрового Двійника необов'язково мати всі ці характеристики, оскільки всі вони пов'язані з інтелектуальним продуктом. Однак деякі з цих характеристик важливі для управління цифровим двійником. З огляду на це найбільш актуальні теми для створення цифрового двійника можна поділити в такий спосіб.

Ідентифікація: продукту потрібна глобальна ідентифікація, щоб зв'язати кожен фізичний продукт з його цифровим представленням.

Для цього можна використовувати такі технології:

- мітки RFID;
- електронний код продукту (EPC);

Він забезпечує унікальну ідентифікацію для кожного фізичного об'єкта в будь-якій точці світу і для всього його життєвого циклу.

Управління даними: дані та інформація про продукт створюються і розвиваються на трьох етапах життєвого циклу продукту: BOL, MOL і EOL.

Розмір даних, що зберігаються на цих етапах, може досягати великих пропорцій. І тут виникає проблема управління великими даними і аналітики для зберігання тільки важливих даних.

Моделі цифрових двійників: на різних етапах життєвого циклу продукту створюються різні типи моделей продуктів.

Деякі з цих моделей:

- системні моделі;
- функціональні моделі;
- тривимірні геометричні моделі;

- супер-фізичні моделі;
- виробничі моделі;
- моделі використання;

Такі моделі іноді можуть взаємодіяти один з одним, і це критична проблема, тому що інформація про цифровий двійник призводить до великих обсягів даних, що генеруються протягом всього терміну служби реального продукту.

Наприклад, інформація про обслуговування і використанні важлива для застосування методів прогнозування поведінки продукту в майбутніх сценаріях.

Людино-комп'ютерний інтерфейс: наявність великої кількості інформації для обробки в різних репозиторіях даних, проблема отримання і відображення необхідної інформації для правильного користувача.

Digital Twin надає інформацію всім користувачам і всім зацікавленим сторонам, залученим в його життя. Але те, як ця інформація буде показана користувачам – це спосіб думати про людино-комп'ютерних інтерфейсів для цифрового двійника.

Комунікація: необхідно мати спосіб комунікації між місцем, де зберігаються дані, і HSI. Цифровий двійник може зберігати цю інформацію в різних базах даних. Однак з появою нових технологій для Інтернету речей (IoT) необхідно мати спосіб отримувати інформацію з цих баз даних і мати інформацію в будь-якому місці і в будь-який момент.

1.2 Огляд існуючих рішень

Як і очікувалося, великі компанії – розробники програмного забезпечення PLM першими запропонували ініціативи цифрових двійників, результатом яких стали реальні послуги. Не дивно, що це ті ж компанії, які наважилися на IoT кілька років тому. Ось кілька компаній, у яких є спеціальні веб-сайти, тематичні дослідження по темі цифрових близнюків, а також реальні продукти, що вже використовуються компаніями по всьому світу:

Таблиця 1.1 Опис існуючих рішень

<i>Назва</i>	<i>Короткий опис</i>
Dassault Systèmes	Основою даної технології є програмне забезпечення, що впроваджує фізичну симуляцію без користувацького інтерфейсу.
Siemens	Розробка – це просунутий фреймворк для побудови бази даних та використання її з метою побудови фізичної симуляції цифрового двійника. Рішення розповсюджується як хмарне рішення
Maplesoft	Продукт являє собою REST API, за допомогою якого можна побудувати просунуту базу даних та симуляцію на їх основі.

1.3 Висновки по розділу

Цифрові двійники – це досить різнобічна технологія, що має перспективу активного використання у різноманітних сферах: від малого діджиталізованого бізнесу, до виробництв та важкої промисловості, для створення повної комплексної симуляції виробничого процесу та контролю за життєвим циклом кінцевого продукту.

Технологія симуляції – це найважливіший крок у автоматизації виробництва та один з найважливіших кроків інтеграції суспільства у IoT. Слід пам'ятати, що технологія симуляції методом цифрового двійника включає не тільки саму симуляцію, а й збір усіх даних та знань про фізичний двійник у єдину базу даних.

Більш того, симуляцію можливо використовувати не лише для обробки даних, а й для управління фізичним двійником.

РОЗДІЛ 2. АНАЛІЗ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Обрана база даних

За визначенням, MongoDB – це база даних з відкритим вихідним кодом, в якій використовується документно-орієнтована модель даних і не структурована мова запитів. На сьогоднішній день це одна з найпотужніших систем і баз даних NoSQL.

Інструмент NoSQL означає, що він не використовує звичайні рядки і стовпці, які ви так часто асоціюєте з керуванням реляційної базою даних. Це архітектура, побудована на колекціях і документах. Базова одиниця даних в цій базі даних складається з набору пар ключ-значення. Це дозволяє документам мати різні поля і структури. Ця база даних використовує формат зберігання документів під назвою BSON, який являє собою двійковий стиль документів JSON. Модель даних є досить високо-еластичною, і дозволяє вам комбінувати і зберігати дані багатовимірних типів без необхідності йти на компроміс з потужними параметрами індексування, правилами запису, доступу і перевірки. Коли ви хочете динамічно змінювати схеми, простою не виникає. Це означає, що ми можемо більше зосередитися на ускладненні роботи з даними, а не витрачати більше часу на підготовку даних для бази даних.

2.2 Архітектура бази даних MongoDB NoSQL

База даних: простими словами, її можна назвати фізичним контейнером для даних. Кожна з баз даних має свій власний набір файлів в файлової системі з декількома базами даних, що існують на одному сервері MongoDB.

Колекція: групу документів бази даних можна назвати колекцією. У СУБД, еквівалентна колекції, є таблицю. Вся колекція існує в єдиній базі даних. Коли справа доходить до колекцій, немає ніяких схем. У середині

колекції різні документи можуть мати різні поля, але в основному документи в колекції призначені для однієї і тієї ж задачі або для досягнення однієї і тієї ж кінцевої мети.

Документ: набір пар ключ-значення може бути позначений як документ. Документи пов'язані з динамічними схемами. Перевага динамічних схем полягає в тому, що документ в одній колекції не обов'язково повинен мати однакову структуру або поля. Крім того, загальні поля в документі колекції можуть містити дані різних типів.

Важливі особливості MongoDB

Запити: БД підтримує спеціальні запити і запити на основі документів.

Підтримка індексації: всі поля документа можна проіндексувати.

Реплікація: БД підтримує реплікацію Master-Slave. MongoDB використовує власний додаток для підтримки декількох копій даних. Запобігання простою бази даних – одна з функцій набору реплік, оскільки він має самовідновлюється сегмент.

Кілька серверів: база даних може працювати на декількох серверах. Дані дублюються, щоб убезпечити систему від помилок в разі відмови обладнання.

Автоматичний розподіл: цей процес розподіляє дані по декільком фізичним розділах. Через сегментування в MongoDB є функція автоматичного балансування навантаження.

MapReduce: підтримує MapReduce і гнучкі інструменти агрегування.

Обробка збоїв: в MongoDB легко впоратися з випадками збоїв. Величезна кількість реплік забезпечує підвищений захист і доступність даних від простоїв бази даних, таких як збої стійки, збої кількох машин, збої центрів обробки даних або навіть мережових розділів.

GridFS: не ускладнюючи стек, можна зберігати файли будь-якого розміру. Функція GridFS розділяє файли на більш дрібні частини і зберігає їх як окремі документи.

База даних без схеми: це база даних без схеми, написана на C ++.

Документно-орієнтоване сховище: воно використовує формат BSON, який є форматом, подібним JSON.

2.3 Приклад схеми документу company у базі даних

Для визначення моделі даних використовувалася мова JSON. Нижче я привів приклад схеми одного із документів. За для більш зручної роботи із БД використовувалась ORM Mongoose, яка дозволяє зручно визначати параметри кожного значення, подібно тому, як це відбувається у SQL базах. Можна побачити, що ми можемо визначити такі параметри як тип, унікальність, обов'язковість поля, а також мінімальну і максимальну довжину кожного поля.

```
name: {
  type: String,
  required: true,
  minlength: 1,
  maxlength: 80
},
photo: {
  type: String,
  required: false,
  minlength: 1,
  maxlength: 596
},
type_id: {
  type: CompanyTypeId,
  ref: 'CompanyType',
  required: true,
  default: 0
},
```

```

activity: {
  type: Number,
  required: true,
  default: 0
},
nip: {
  type: Number,
  required: true,
  default: 0
},
user_id: {
  type: String,
  required: true,
}

```

Таким чином ми отримуємо досить гнучку структуру, що ідеально підходить для адаптивних та гнучких систем на кшталт такої, яку я розробляю.

2.4 Огляд технологій серверної частини додатку

Node.js – це серверна платформа, побудована на основі Google Chrome JavaScript Engine (V8 Engine). Node.js був розроблений Райаном Далем в 2009 році, ми будемо використовувати – v0.10.36. Визначення Node.js в його офіційній документації виглядає наступним чином:

Node.js - це платформа, побудована на середовищі виконання JavaScript Chrome для простого створення швидких і масштабованих мережевих додатків. Node.js використовує керовану подіями модель введення-виведення, яка робить його легким і ефективним, що ідеально відповідає вимогам інструменту розробки додатків з інтенсивним використанням даних в реальному часі, які працюють на розподілених пристроях.

Node.js – це кроссплатформена середовище з відкритим вихідним кодом для розробки серверних і мережевих додатків. Додатки Node.js написані на JavaScript і можуть запускатися в середовищі виконання Node.js на наступних операційних системах:

- OS X;
- Microsoft Windows;
- Linux;

Node.js також надає багату бібліотеку різних модулів JavaScript, яка в значній мірі спрощує розробку веб-додатків з використанням Node.js.

Нижче наведені деякі з важливих функцій, які роблять Node.js кращим вибором для поточного продукту.

- Асинхронність і керованість подіями – всі API бібліотеки Node.js є асинхронними. Це означає, що сервер на основі Node.js ніколи не чекає, поки API поверне дані. Сервер переходить до наступного API після його виклику, і механізм повідомлення подій Node.js допомагає серверу отримати відповідь від попереднього виклику API.
- Швидкість. Бібліотека Node.js, створена на базі технології Google Chrome V8 JavaScript Engine, дуже швидко виконує код.
- Однопоточний, але добре масштабується – Node.js використовує однопоточну модель з циклом подій. Механізм подій допомагає серверу реагувати неблокуючим чином і забезпечує високу масштабованість сервера на відміну від традиційних серверів, які створюють обмежені потоки для обробки запитів. Node.js використовує однопоточну програму, і одна і та ж програма може обслуговувати набагато більшу кількість запитів, ніж традиційні сервери, такі як HTTP-сервер Apache.
- Без буферизації – додатки Node.js ніколи не буферизують дані. Ці додатки просто виводять дані по частинах.
- Ліцензія – Node.js доступний за ліцензією MIT, що значить економію ресурсів при розробці.

2.5 Огляд технологій користувацького інтерфейсу

WebGL – це інтерфейс програмування графічних додатків, створений для використання в веб-додатках. Він заснований на вбудованому стандарті (ES) відкритої графічної мови (OpenGL).

WebGL використовується розробниками для забезпечення незалежних від платформи засобів створення інтерактивних графічних додатків в мережі. WebGL використовується не тільки для малювання графіки в 2D і 3D, а й для прискорення роботи веб-редакторів зображень і їх ефектів, а також для моделювання фізики.

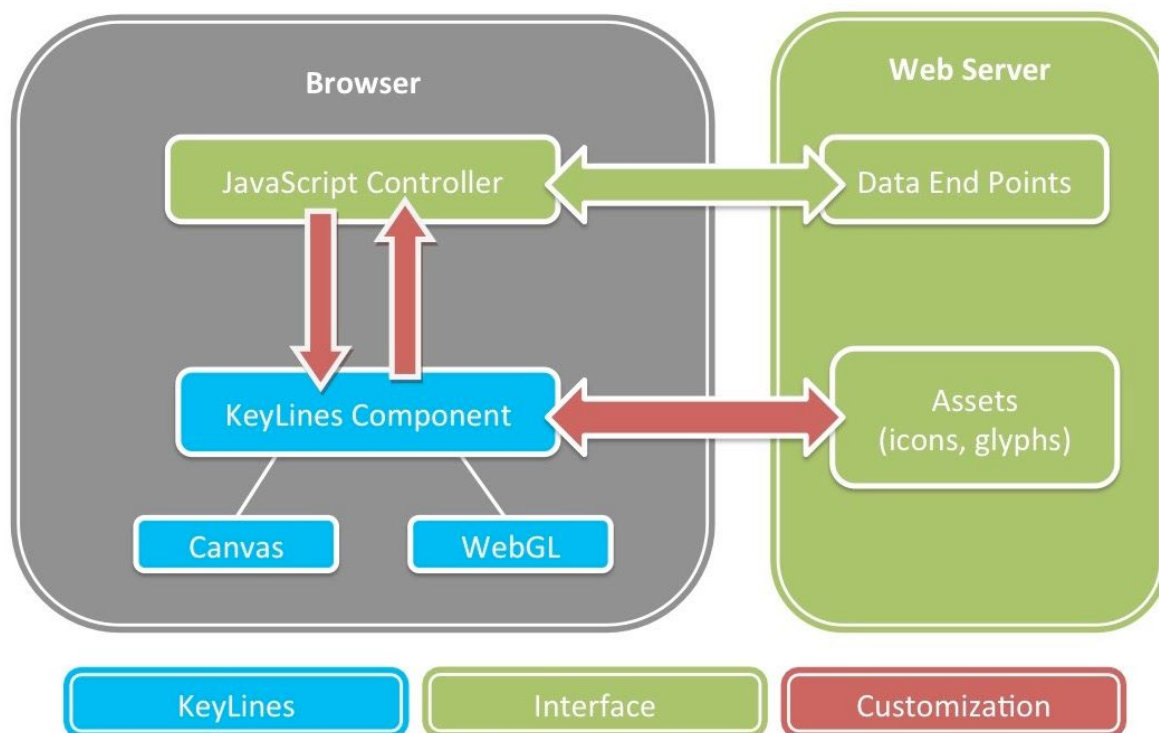


Рисунок 2.1 Схема роботи WebGL

Хоча WebGL функціонально заснований на OpenGL ES, він частково написаний на JavaScript. WebGL використовується для візуалізації інтерактивної 2D- і 3D-графіки в сумісних веб-браузерах. API дозволяє користувачам переглядати інтерактивний контент на веб-сторінках з прискоренням графічного процесора без попереднього завантаження або

установки будь-яких модулів. Для розробників WebGL забезпечує низькорівневий доступ до обладнання зі знайомою структурою коду OpenGL ES.

WebGL спочатку був створений Mozilla. В даний час API розроблений і підтримується некомерційною організацією Khronos Group. Код зібрано з двох частин: керуючий код і код шейдера. Керуючий код створено на JavaScript і обробляється центральним процесором. Код шейдера, написаний на OpenGL ES, який має багато спільного з такими мовами, як C або C ++, і обробляється на GPU.

2.6 Огляд технології для управління розгортанням

Docker – це відкрита платформа для розробки, доставки та запуску додатків. Docker дозволяє відокремити програми від інфраструктури, для того щоб швидко доставити програмне забезпечення. За допомогою Docker можна керувати інфраструктурою так само, як і програмами. Скориставшись методологіями Docker для швидкої доставки, тестування та розгортання коду, ми зможемо значно скоротити затримку між написанням коду та його запуском у виробництво.

Docker надає можливість упаковувати та запускати додаток у вільно ізолюваному середовищі, яке називається контейнером. Ізоляція та безпека дозволяють запускати багато контейнерів одночасно на одному сервері. Контейнери легкі, оскільки їм не потрібно додаткове навантаження гіпервізора, але вони працюють безпосередньо в ядрі хост-машини. Це означає, що можна запускати більше контейнерів на певній апаратній комбінації, ніж якби використовувались віртуальні машини. Docker пропонує інструменти та платформу для управління життєвим циклом контейнерів:

- Розробка програми та допоміжних компонентів за допомогою контейнерів.

- Контейнер стає пристроєм для розповсюдження та тестування програми.
- Розгортання програми у виробничому середовищі, як контейнер або організовану службу. Це працює однаково, незалежно від того, чи є виробниче середовище локальним центром обробки даних, хмарним постачальником чи гібридом двох.

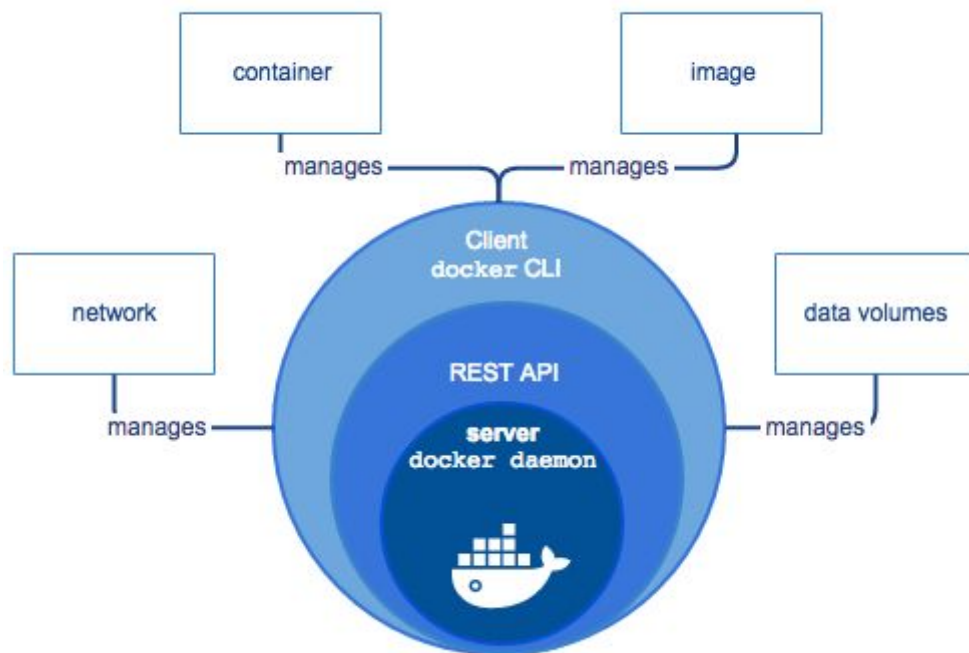


Рисунок 2.2 Структура Docker

Розглянемо можливості, які можна отримати завдяки технології розгортання.

Швидка, послідовна доставка програм:

Docker впорядковує життєвий цикл розробки, дозволяючи розробникам працювати в стандартизованих середовищах, використовуючи локальні контейнери, що надають програми та послуги. Контейнери чудово підходять для безперервної інтеграції та безперервної доставки робочих процесів (CI / CD).

Адаптивне розгортання та масштабування:

Контейнерна платформа Docker забезпечує надзвичайно портативні робочі навантаження. Контейнери Docker можуть працювати на локальному ноутбукі розробника, на фізичних або віртуальних машинах в центрі обробки даних, у хмарних провайдерах або в різних середовищах.

Переносимість і легкий характер Docker також спрощують динамічне управління робочими навантаженнями, масштабування або знос додатків та послуг відповідно до потреб бізнесу, майже в реальному часі.

Запуск більшої кількості робочих навантажень на тому самому обладнанні:

Docker забезпечує життєздатну, економічно вигідну альтернативу віртуальним машинам на базі гіпервізора. Docker ідеально підходить для середовищ із високою щільністю, а також для малих та середніх розгортань, де потрібно отримати найбільш високий рівень оптимізації навантажень на програмне забезпечення та сервер.

2.7 Висновки до розділу

Як й було описано у розділі, у проєкті використовуються лише розповсюдженні технології розробки. Таке рішення було прийнято для того, щоб забезпечити проєкт високим рівнем підтримки із боку розробників, адже при використанні нерозповсюджених технологій, вартість розробки може значно зрости.

Крім того, варто зазначити, що технології для серверної і клієнтської частини працюють на основі мови програмування JavaScript.

РОЗДІЛ 3. РОЗРОБКА АЛГОРИТМІЧНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Алгоритм визначення структури системи симуляції цифрового двійника

Моделі і дані – це ядро цифрового двійника. Однак створення віртуальних моделей – складний і спеціалізований процес, як і об'єднання і аналіз даних. Для користувачів, що не володіють відповідними знаннями, створити і використовувати цифрового двійника складно.

Отже, абсолютно необхідно, щоб моделі могли використовуватися спільно користувачами, а аналіз даних передавався стороннім професійним організаціям. Більш того, в контексті усупільнення виробництва фізичні ресурси, задіяні у виробництві, географічно розподілені.

Завдяки характеристикам використання на вимогу динамічної реконфігурації і незалежності від платформи сервіси відкривають шлях до вирішення проблем, згаданих вище. Через Інтернет користувачі можуть отримувати доступ до різних елементів і використовувати їх для створення цифрового двійника.

Інкапсуляція сервісу призначена для перекладу різних компонентів цифрового двійника в сервіси з єдиним описом. Першим і найбільш важливим кроком інкапсуляції послуги є створення інформаційного шаблону, який складається з різноманітної інформації.

Для фізичних об'єктів ця інформація включає в себе:

- основні атрибути (ім'я, ідентифікатор, адреса, тощо)
- QoS (час, вартість, надійність, задоволеність, тощо)
- можливості (точність, розмір, процес, тощо)
- стан в реальному часі (перевантаження, простой, необхідність обслуговування, тощо)

А також вхід і вихід. Інформаційний шаблон фізичного об'єкта можна описати формулою:

$$PO = \{Basic, QoS, Cap, Status, Input, Output\}.$$

Де Basic позначає основні атрибути для ідентифікації фізичної об'єкта.

QoS (якість обслуговування) означає оцінку продуктивності фізичного об'єкта для проведення вибору користувачем.

Cap позначає функції, які може виконувати фізичний об'єкт.

Status вказує, чи доступний фізичний об'єкт. Як приклад, на верстаті з ЧПУ, виробнича потужність включає в себе:

- можливість обробки форми (площини, отвори, різьблення, тощо)
- точності (шорсткості, прогину, тощо)
- обробки деталей (зубчастих коліс, пластин, тощо)
- розмір (висота, ширина, координати, тощо)

Status включає:

- перевантаження
- холостий хід
- обслуговування

Для введення (матеріалу у ЧПУ можуть використовуватися такі матеріали, як сталь, алюмінієвий сплав, тощо). А на виході може бути деталь або невеликий продукт.

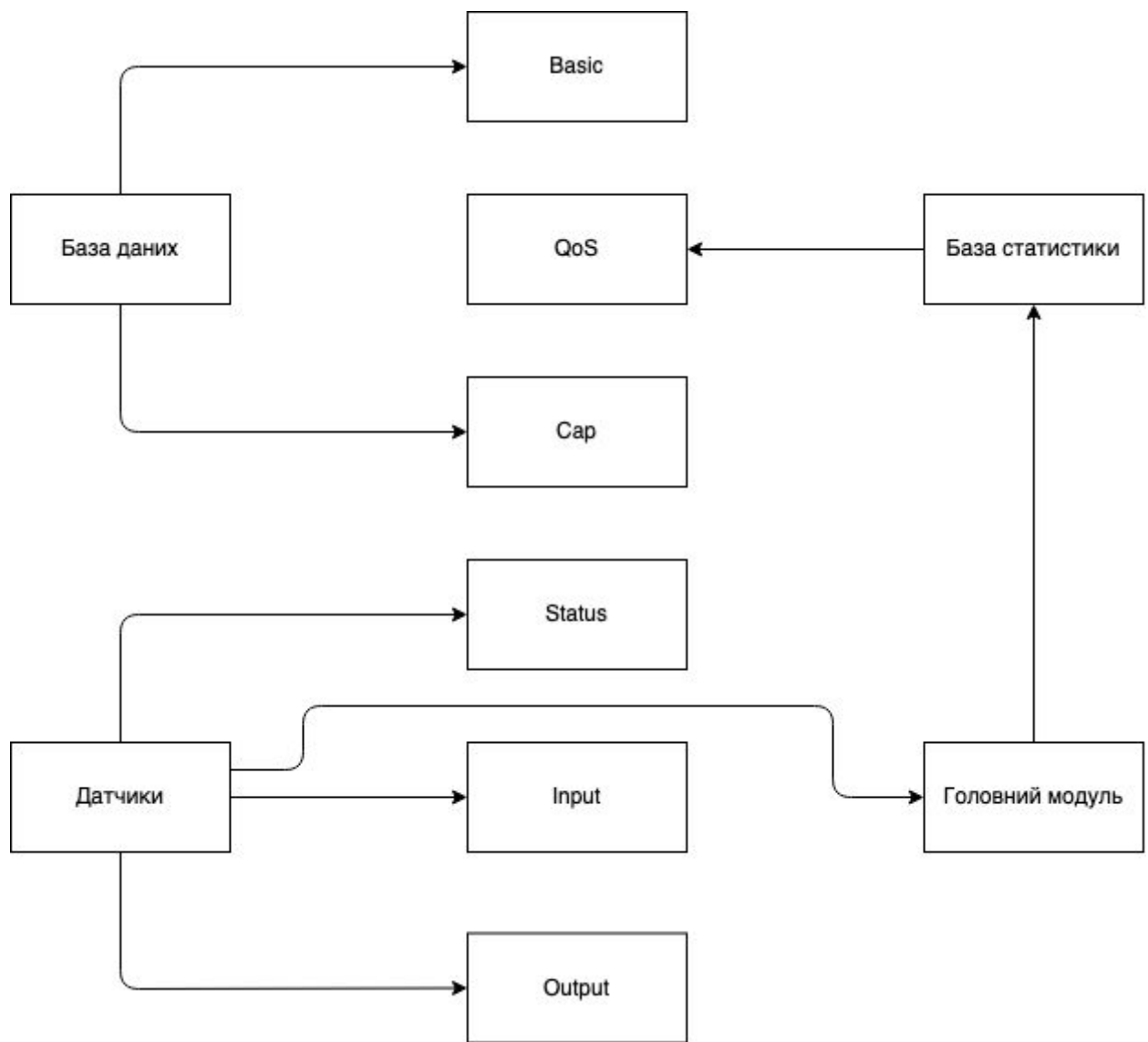


Рисунок 3.1 Схема сутностей

Кожний із цих параметрів можна сформулювати як:

$Basic = \{type, name, ID, address, color, delivery_time, \dots\},$

$QoS = \{cost, time, reliability, trust, Fun_similarity, \dots\},$

$Cap = \{shape_cap, precision_cap, parts_cap, size_cap, \dots\},$

$Status = \{load, health, \dots\}.$

Аналогічним чином віртуальні моделі можуть бути описані за допомогою інформаційного шаблону. На відміну від фізичних об'єктів, віртуальні моделі можуть використовуватися декількома користувачами одночасно. Крім того, віртуальні моделі можна копіювати без повторного створення для одного і того ж, або рівного фізичного об'єкта. Отже, коли віртуальні моделі інкапсулюються в сервіси для спільного використання

користувачами, переваги можуть отримати не тільки творці моделей, але і користувачі можуть скоротити витрати і час. Інформаційний шаблон віртуальної моделі можна описати як:

$$VM = \{Ori_Phy, Creator, Ori_ID, Cup_ID, Owner, QoS, Online_Site, Input, Output, \dots\}.$$

Де *Ori_phy* позначає вихідний фізичний об'єкт, з якого створюється віртуальна модель. Творець – це той, хто будує модель на основі своїх спеціальних знань. *Ori_ID* – це вихідний ідентифікаційний номер при першому створенні моделі. *Cup_ID* – це ідентифікаційний номер поточної копії. Власник – той, хто володіє моделлю. Власник, який володіє авторськими правами може бути творцем, також він може отримувати прибуток, здаючи в оренду моделі або продаючи копії.

Подібно фізичному об'єкту, *QoS* позначає оцінку продуктивності моделей, включаючи вартість, надійність, функції, тощо. *Online_site* позначає онлайн-адресу, за якою користувачі можуть отримати доступ до моделей або завантажити їх. Вхід і вихід можуть відрізнятися в залежності від конкретних моделей. Ми будемо використовувати для цього згенеровану адресу API, для взаємодії із даними і БД.

Крім того, дані дуже важливі для інтелектуального виробництва, однак через різність стандартів, протоколів та інтерфейсів зв'язку, дані важко отримати і зрозуміти. Використовуючи єдиний шаблон для опису даних, користувачі можуть зручно використовувати дані. Як правило, інформація про дані включає в себе постачальник даних, якому належать дані, джерела даних, в яких збираються дані, ідентифікатор даних, який використовується для ідентифікації даних, тип даних, що визначає тип даних, і реферат даних, який представляє собою коротку інструкцію для позначення даних, подібно тому, як це декларують розповсюдженні транспортні протоколи. Отже, можна описати як:

$$Data = \{D_prov, D_source, D_ID, D_type, D_abstract, \dots\}.$$

Інформаційні шаблони описують велику частину інформації, яку користувачі хочуть і повинні знати. Після того, як інформаційні шаблони встановлені, різні види фізичних об'єктів, віртуальних моделей і даних інкапсулюються в сервіси на основі мови опису сервісів, такої як мова XML. Наприклад, інформаційний шаблон штампувального верстата і блоку управління ПЛК інкапсулюється службою на основі XML. При цьому інформаційний шаблон включає в себе основну інформацію, можливості обробки, параметри процесу і інформацію про обсяг штампувального верстата і блоку управління ПЛК.

3.2 Серверний додаток цифрового двійника

Після інкапсуляції сервісів цифрових двійників, вони публікуються в пулі сервісів і на платформі управління, де вони адаптуються для спільного використання різними користувачами.

Служби цифрових двійників складаються із наступних служб:

- служба обладнання;
- технологічна служба;
- служба тестування;
- служба даних;
- служба знань;
- служба алгоритмів;
- служба моделей;
- служба моделювання;

Крім того, існує безліч допоміжних служб, таких як:

- фінансові послуги;
- послуги логістики;
- послуги навчання;
- послуги з ремонту обладнання;

Управління послугами включає в себе:

- пошук;

- зіставлення;
- планування;
- комбінацію;
- транзакцію;
- відмовостійкість;

Завдання відправляється на платформу управління за допомогою стандартного HTTP POST запиту. Потім він розбивається на підзадачі, які можуть бути виконані за допомогою однієї служби. Згідно QoS виконується узгодження попиту і пропозиції виробничих послуг і планування для вибору оптимальних послуг. Після транзакції вибрані служби викликаються і об'єднуються для спільного виконання завдання. Нарешті, результати повертаються користувачам.

Сервіси цифрового двійника можуть використовуватися при проектуванні продуктів, плануванні виробництва, виконанні виробництва і РНМ (прогнозування і управління здоров'ям) обладнання.

У дизайні продукту – це процес двосторонньої взаємодії між очікуваним і фізичним світом. Дизайн, керований цифровим двійником, повинен перетворити очікуваний продукт в розумі дизайнера в цифрове представлення в інтерпретуємому світі на основі існуючих фізичних продуктів.

Щоб створити новаторський продукт, дизайнер повинен вивчити безліч даних, щоб отримати цінні знання. Однак дані про продукт – один з найважливіших активів, до якого нелегко отримати доступ. Крім того, у дизайнерів немає професійних навичок для обробки великих обсягів даних.

Сервіс – це відповідь на вищевказані проблеми. Дизайнери просто відправляють свої потреби в платформу управління послугами. Менеджери служб будуть відповідати службам даних, які потрібні розробникам, і службам моделей і алгоритмів, які використовуються для обробки даних.

Завдяки викликам, об'єднанню і роботі цих служб результати повертаються дизайнерам. Крім того, після розробки функціональної

структури і компонентів продукту необхідно перевірити якість і здійсненність дизайну.



Рисунок 3.2 Принцип роботи системи управління

Завдяки цифровому двійникові дизайнери можуть швидко і легко прогнозувати поведінку продукту за допомогою перевірки віртуальних продуктів, не чекаючи створення прототипу продукту. Але для віртуальної перевірки потрібні моделі виробничого майданчика (наприклад, виробничої лінії або цеху), яких у проектувальників немає. Після інкапсуляції сервісів та моделей, їх можна використовувати за допомогою пошуку, зіставлення, планування і виклику сервісів. За допомогою послуг, цифровий двійник може бути легко використаний в дизайні продукту, що може зробити дизайн продукту більш ефективним, зменшити невідповідності очікуваної поведінки і поведінки дизайну, а також значно скоротити цикли проектування та знизити витрати.

Загалом, виробництво продукції – це весь процес від введення сировини до випуску готової продукції, який виконується в цеху. Щоб скоротити витрати, час виробництва і підвищити ефективність, необхідно планувати виробництво з приреченням виробничого процесу. На етапі

планування виробництва і виконання виробництва цифровий двійник забезпечує ефективний метод для складання плану, оптимізації та виконання процесу.

По-перше, виробнича задача відправляється на платформу управління послугами, а також узгоджує пропозицію і попит на ресурсні послуги. Потім на основі стану фізичних ресурсів (наприклад верстатів або маніпуляторів) у реальному часі складається виробничий план. Цифровий двійник в цеху може моделювати плани в віртуальному просторі і виявляти потенційні конфлікти ще до самого фактичного виробничого процесу. Однак створення цифрового двійника в цеху – це складна і спеціалізована робота, яку необхідно спланувати з досить великою точністю, особливо моделі, що включають геометрію, правила, поведінку і динамічні моделі.

За допомогою сервісів ці моделі не обов'язково створювати самим виробником. Для фізичного обладнання та загальних правил їх моделювання, розробку можна купити для використання у вигляді послуг від незалежних об'єднань розробників або комерційних компаній. Чинному виробнику досить створити спеціальні моделі, які підходять тільки їм самим. Крім того, під час роботи цеху деякі служби, такі як обробка даних, управління цехом, тощо, необхідно викликати з системи служб цифрового двійника цеху, або виконувати на головному сервері.

Зниження продуктивності фізичного обладнання неминуче. Коли обладнання виходить з ладу, це може призвести до високих витрат на технічне обслуговування і відстрочення виконання завдань.

РНМ необхідний для моніторингу стану обладнання, прогнозування та діагностики несправностей устаткування і терміну служби компонентів. У РНМ, керованому цифровим двійником, віртуальні моделі фізичного обладнання синхронізуються з реальним станом обладнання. Стан роботи обладнання і стан компонентів відслідковуються в режимі реального часу.

Цифрове дзеркало високої якості для обладнання забезпечує доступ до обладнання навіть далеко від нього. Крім того, взаємодія цифрового

двійника може зменшити перешкоди від зовнішнього середовища, підвищуючи точність. У наведеному вище процесі доступ до моделей здійснюється через служби. Більш того, при виникненні несправностей викликаються ремонтні служби для ремонту або заміни вийшов з ладу обладнання.

3.3 Фізична частина процесу інтеграції

В контексті оцифровки даних збір, зберігання і аналіз даних будуть набувати все більшого значення. ІТ-середовища на підприємствах, починаючи від вбудованих систем на рівні цеху і закінчуючи системами управління виробництвом і виробничими процесами або системами планування ресурсів, стануть основою для цифрового двійника виробничих підприємств.

Однак, особливо в рамках малого і середнього бізнесу, ІТ-системи впроваджуються лише частково, зокрема тому що відсутні цілісні рішення. Отже, необхідно розробити і впровадити концепцію навчання, що дозволяє учасникам ознайомитися з необхідними технологіями, зусиллями по впровадженню і перевагами цифрового двійника.

Більш того, основна задача у проблемі оцифрування даних полягає у розробці низки універсальних датчиків, що не потребує великої кількості коштів для встановлення.

Рішення цієї задачі полягає у створенні єдиного інтерфейсу для обміну даними, та створені API для можливості створювати незалежне програмне забезпечення сторонніми розробниками.

Найменшою одиницею системи є датчик, основна задача якого полягає у зборі елементарних даних, обробки даних до стандартного формату та відправці даних до модулю Wi-Fi за допомогою внутрішнього модулю Wi-Fi.



Рисунок 3.3 Алгоритм циркуляції даних у датчику

Збір даних є важливою частиною створення цифрового двійника виробничих середовищ. Що стосується обробки даних в промисловому виробництві, можна виділити два типи даних, а саме, енергозалежні і енергонезалежні дані.

Останній фіксується вимірами і інтерв'ю, так як він складається з відповідних макетів, специфікацій обладнання, документів, включаючи зміни, відомості матеріалів і рівні кваліфікації співробітників.

Для збору мінливих даних, включаючи рух співробітників, рух коштів виробництва, потік матеріалу, час виробництва, а також час обробки і використовувану потужність обладнання, використовуються системи визначення місця розташування в реальному часі і системи обробки зображень, такі як нейронної мережі для розпізнавання образів. Для визначення місця розташування в приміщенні в реальному часі реалізована бездротова система, що складається з якорів і датчиків на основі радіозв'язку. Обробка зображень виконується на спеціальній системі на базі операційної системи з відкритим вихідним кодом Linux.

Для обробки зображень прийнято використовувати нейромережі. У даного підходу є великий плюс, якщо роздивляться його у розрізі виробничої компанії, адже виробництво – це незмінний, періодичний процес, що дозволяє нам отримати велику базу зображень для тренування нейромережі.

Така база називається Data Set та представляє собою розмічений набір даних, наприклад зображень готової деталі, та дефектної деталі, який використовується для тренування нейромережі, та подальшого її тестування, для визначення точності детекції навченої нейромережі.



Рисунок 3.4 Принцип передачі даних із фізичного двійника

Його мета – надати рішення для збору даних з комбінацією існуючих технологій. Крім того, учасники повинні знати класифікацію даних, необхідних для цифрового двійника, і їх зміст.

3.4 Локальний препроцесінг даних

Після збору даних необхідна концепція зберігання даних, включаючи інтерфейси. Оскільки цифровий двійник, заснований на стандартизованих і

усталених технологіях і рішеннях, потенційно демонструє більш широкий спектр застосування, представлено існуюче хмарне рішення для зберігання даних. Крім того, основною вимогою для збору і зберігання даних є мережевий інтерфейс між технологіями збору даних і хмарним рішенням для зберігання даних.

Основна схема обміну даними представлена на наступній діаграмі:



Рисунок 3.5 Архітектура без урахування ризиків

Мета – надати рішення для зберігання даних, засноване на існуючих хмарних рішеннях. Крім того, учасники повинні знати про існування єдиного хмарного репозиторія з даними.

Щоб використовувати зібрані і збережені дані, необхідна система, що містить програмне забезпечення, а також необхідні інтерфейси. Крім того, необхідно створити таку архітектуру, щоб виділити мікросервіси під кожну окрему задачу.

Як і показано на діаграмі вище, ми відокремили Контролер від симуляції, адже оперування даними та визначення бізнес-логіки не є основною задачею для симуляції методом цифрового двійника.

Але у даній схемі існує ризик відказу модулю препроцесінгу, що веде за собою збій у роботі системи. Більш того, при відказі модулю симуляції виникає та ж сама проблема. Для вирішення цієї ситуації було вирішено додати модуль бекапування вхідних даних:

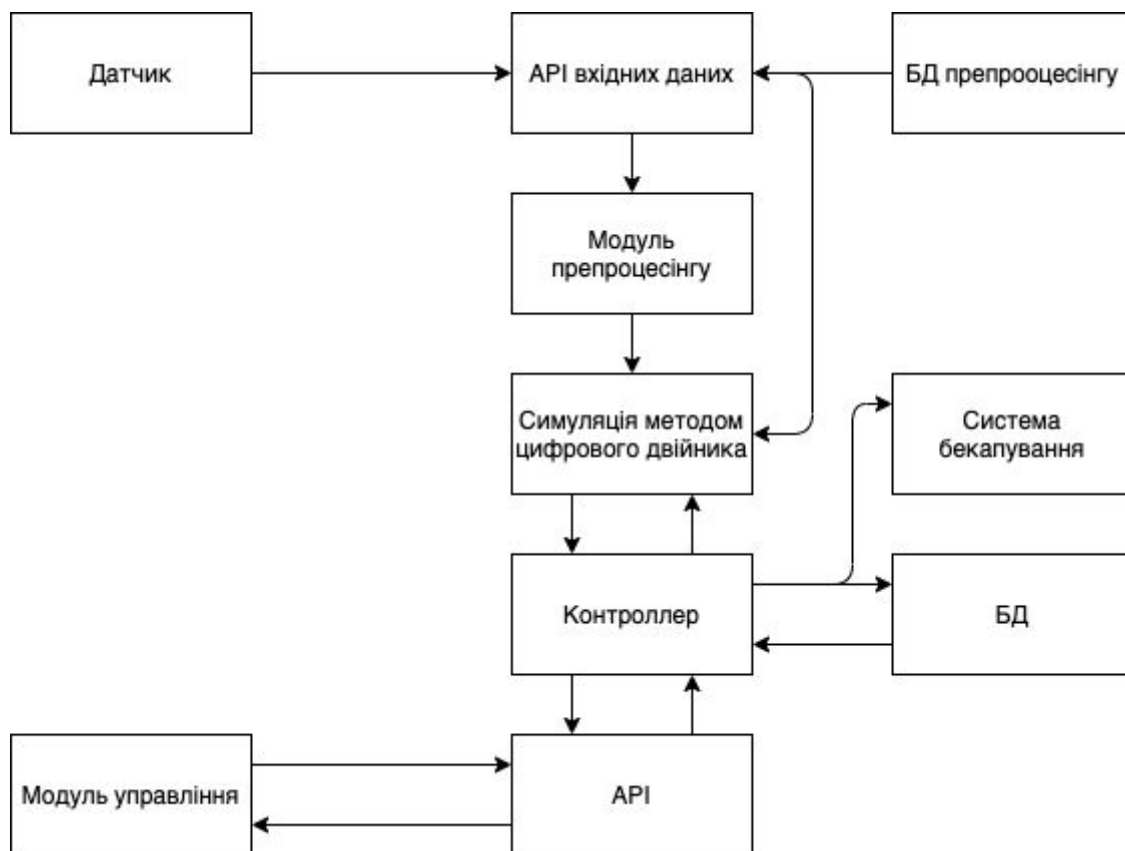


Рисунок 3.6 Архітектура із урахуванням ризику

3.5 Інтеграція робототехнічних модулів із симуляцією

Існує чотири основні підкомпоненти, які розгортаються в рамках запропонованої інфраструктури цифрового двійника:

- а) Менеджер ресурсів;
- б) Менеджер датчиків;
- в) Менеджер макета;
- г) Конструктор середовища 3D;

Ці програмні компоненти були розроблені на основі принципів ROS, забезпечуючи зв'язок між мережевими вузлами та швидку інтеграцію з існуючими робототехнічними програмами.

Зокрема, описана інфраструктура Digital Twin була розгорнута та протестована на ПК, що працює на Ubuntu 14.04 з версією ROS indigo. Кожен із включених компонентів реалізований як вузли Node.js, що забезпечують ROS-інтерфейси (теми, послуги, дії) для обміну відповідною інформацією.

Файли конфігурації, які обробляються та генеруються з Digital Twin, відповідають стандартним правилам спільноти ROS. Менеджер ресурсів відповідає за реєстрацію в Digital Twin будь-якого нового ресурсу, представленого в магазині.

Заснований на уніфікованій моделі даних ресурсів, Менеджер ресурсів зберігає всі атрибути, що складають модель ресурсу, такі як максимальне корисне навантаження, мінімальна швидкість, розташування, конфігурація перетворення (.urdf), конфігурація шляху (.yaml) і руху (.srdf) тощо.

Два підкомпоненти, а саме моніторинг розташування ресурсів та моніторинг статусу ресурсу, відповідають за моніторинг стану та місцезнаходження кожного мобільного ресурсу в режимі реального часу та оновлення фактичних значень на сервері Digital Twin.

Процес починається із того, що ініціюються конкретні теми та послуги ROS для кожного передплаченого ресурсу, що транслює свої дані, пов'язані з реальним часом.

Sensor Manager відповідає за взаємодію з драйверами ROS існуючих датчиків та реєстрацію їх даних конфігурації у сховищі Digital Twin із використанням уніфікованого формату моделі даних. Усі сенсорні дані стають доступними для планувальників роботів, використовуючи шаблон публікації/передплати як механізм зв'язку. Датчики однозначно ідентифікуються за їх ідентифікаторами, підтримуючи список відповідних тем.

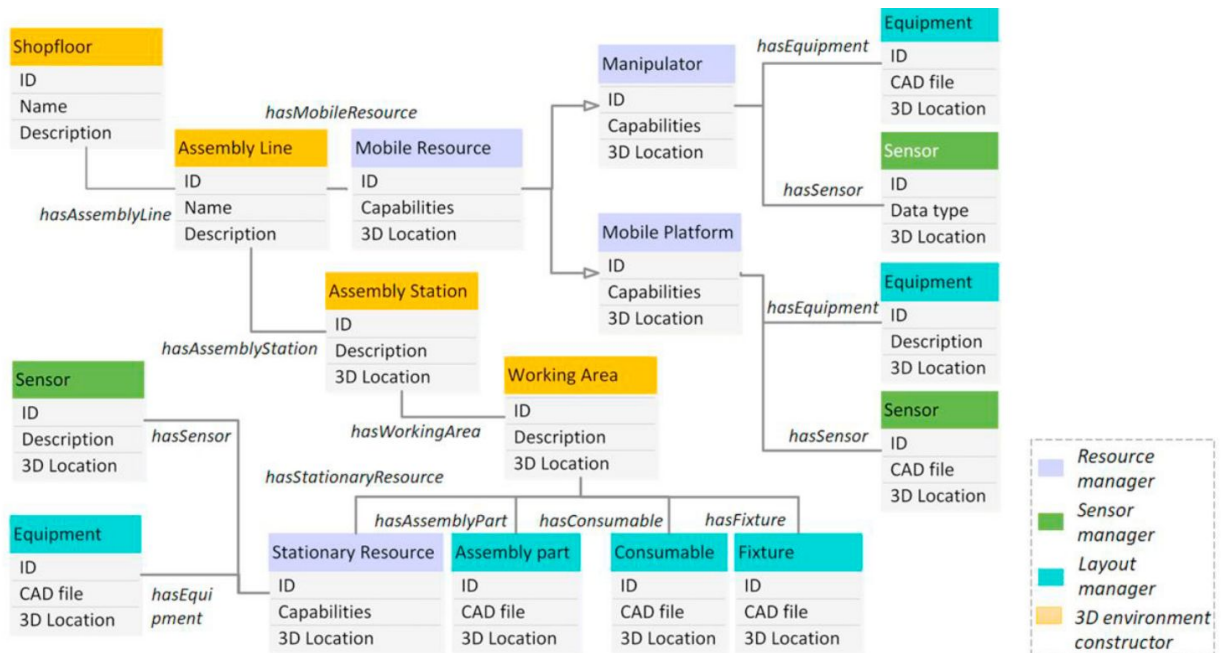


Рисунок 3.7 UML діаграма структур даних

Для полегшення використання стандартних алгоритмів планування руху та траєкторії (наприклад, gmapping, amcl, ompl) 2D-3D комбінований модуль даних даних дозволяє легко та динамічно об'єднати кілька даних датчика в одну тему.

Для ефективного представлення всієї площі статичний макет повинен бути представлений всередині Digital Twin. Диспетчер макетів відповідає за контроль та зберігання всіх файлів САПР, пов'язаних зі статичними кріпленнями, деталями та виробами у форматі .sdf, визначаючи також зіткнення, інерцію та параметри маси.

Цей компонент дозволяє користувачеві завантажувати файл САПР і конфігурувати різні параметри, що стосуються:

- а) частин, що беруть участь у процесі;
- б) стаціонарних приладів, що входять до складу виробництва;

Кінцевим компонентом ланцюжка процесів є конструктор 3D-середовища. Цей компонент отримує розташування всіх деталей, приладів, датчиків та ресурсів для побудови середовища з глобальним світовим каркасом. Окрім статичних частин, положення яких визначається

на фазі конфігурації, є також рухомі об'єкти та перешкоди, які не є добре зафіксованими і їх потрібно визначити під час виконання.

Таким чином, конструктор середовища 3D надає інтерфейс з ROS Services та Topic для відстеження та оновлення положення всіх деталей всередині цеху на основі оновлень в реальному часі, що надходять від датчиків підлоги виробництва.

Було проведено багато досліджень, пов'язаних з уникненням зіткнень між ресурсами та некартованих перешкод усередині приміщення виробництва. Використовуючи існуючі алгоритми, Digital Twin надає інтерфейси із планувальниками шляхів і траєкторій роботів для досягнення перепланування на сервері на основі отриманої реальної інформації з підлоги виробництва. Впроваджений інтерфейс компонента планування шляху заснований на стеці навігації ROS для мобільних роботів. Зокрема, Digital Twin забезпечує:

- а) перетворення для кожного координатного кадру робота, що описано у файлі .urdf;

- б) датчики відстані та інформацію про одометрію, необхідні для картографування та локалізації ресурсу, а також для уникнення зіткнень;

- в) глобальна карта виробництва - 3D-карта, створена в сітці заповнення за допомогою бібліотеки OctoMap та карти витрат з інформацією про перешкоди;

- д) конфігурація планувача. Для планування руху роботизованих систем модель Digital Twin забезпечує інтерфейс з MoveIt. Для налаштування планування руху потрібні три види інформації: а) файл urdf робота, б) файл .srdf робота в) MoveIt! файли конфігурації для спільних обмежень, кінематики, планування руху, введення сприйняття та іншої інформації;

Для забезпечення траєкторії руху робота без зіткнень Digital Twin забезпечує сцену глобального планування всіма об'єктами та ресурсами. Ця сцена планування публікується як карта зайнятості, побудована

конструктором 3D-середовища. Потім ця карта оновлюється в режимі он-лайн з використанням наявних мультисенсорних даних.

Запропонована система була частково впроваджена та випробувана на прикладі з сектору виробництва вогнегасників для штампування донної пластини вогнегасника. Для ефективного виконання сценарію MRP повинен сприймати:

- етап роботи станку
- температуру проссу
- кількість доступних листів металу
- якість деталі для визначення ймовірності отримання браку
- заповненість контейнеру із готовими виробами
- навантаження на електрогенератор
- наявність людини

3.5 Висновки по розділу

Хмарні технології все більше розповсюджуються у світі інформаційних технологій за рахунок своєї адаптивності та гнучкості. Найбільшу користь хмарні технології приносять у сфери із розподіленими системами на кшталт виробництва.

Архітектура, яка була впроваджена та описана у розділі 3, дозволяє використовувати хмарні технології для створення сервісу як послуги обробку даних та симуляції даних.

Завдяку тому, що архітектура пропонує стандартизоване API для роботи із даними та система має механізм для персоналізації – єдиний код та єдиний продукт можна використовувати практично без змін у багатьох підприємствах.

Головна задача проекту – надати більш дешеву та гнучку альтернативу існуючим рішенням. Гнучкість досягається за рахунок системи датчиків, а оптимізація ціни – за рахунок стандартизованого

програмного забезпечення, тому, можна вважати, що свої цілі данна архітектура виконує.

РОЗДІЛ 4. МАРКЕТИНГОВИЙ АНАЛІЗ СТАРТАП-ПРОЕКТУ

Основною задачею стартапу є швидкий вихід на ринок та отримання коштів від монетизації свого рішення, для подолання цього шляху стартап повинен пройти низку кроків, які визначаються від моделі та типу продукту, на основі котрого створюється стартап.

Для будь-якого стартап-проекту найважливішим є успішний ринковий старт. Фахівці зі створення та розвитку стартап-проектів окремо відзначають, що відсутність маркетингових знань та умінь, що уможливляють розробку ринково затребуваного проекту із вихідної ідеї, є основною причиною високого рівня банкрутств стартап-компаній, і ця проблема може бути вирішена за рахунок навчання винахідників. Відповідно, основним призначенням даного розділу є отримання знань щодо суті, основних принципів розроблення стратегії ринкового впровадження та маркетингового управління інноваційними стартап-проектами у промислових галузях економіки, використання ефективних маркетингових інструментів просування високотехнологічних продуктів виробництва та послуг .

4.1 Опис ідеї проекту

Ідея проекту полягає у створенні програмного забезпечення з відкритим ісходним кодом та датчиків для створення цифрового двійника на базі виробничих підприємств.

Фактично, ми гарантуємо дешеву альтернативу для швидкого впровадження технологічного рішення на підприємстві, що дозволить бізнесу отримати більш якісний результат за менші кошти за рахунок стандартизації життєвого циклу процесу впровадження.

На даний момент, симуляція методом цифрового двійника недоступна без інвестування великої кількості коштів, тому що процес впровадження технології не є стандартизованим, через що присутня велика кількість додаткових етапів впровадження.

Головною особливістю даного стартап-проекту є його адаптивність та гнучкість. Замість того, щоб створювати унікальне рішення під кожно виробництво ми надамо велику кількість модулів із яких можна буде зібрати необхідну систему.

Крім модулів, ми також запропонуємо виробникам стандартизовані інтерфейси для інтеграції модулів у єдину систему. Фактично проект являє собою SaaS-рішення для інтеграції модулів власного виробництва. Особливість у тому, що розробка платформи для з'єднання модулів не буде регулюватися з боку розробника, що допускає створення альтернативних SaaS рішень, що лише підсилить позиції датчиків власного виробництва на ринку. Адже таке рішення забезпечить конкуренцію, що позитивно відобразиться на ринку та якості продуктів у цілому. Адже конкуренція передбачає постійний пошук нових клієнтів та покращення якості продукту.

Якщо говорити про пошук нових клієнтів, то це не лише стратегія отримання клієнтів у конкурента, а й пошук нових клієнтів у суміжних сферах.

Таблиця 4.1.1 Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Створення програмного забезпечення для фізичної симуляції та датчиків для збору даних.	1. виробництво та промисловість; 2. транспорт; 3. сортувальні системи для логістики; 4. логістичні компанії; 5. військові системи (системи маніпулювання об'єктами з небезпечних матеріалів);	Економічні

Таблиця 4.1.2 Визначення сильних, слабких а нейральних сторін

№	Техніко-економ ічні характеристик и ідеї	Потенційні товари/концепції конкурентів			W (слабка сторон а)	N (нейт- ральна сторон а)	S (сильна сторон а)
		Мій проект	Конкур ент 1	Конкур ент 2			
1.	Вартість впровадження	Програ мна та фізична база для симуля ції Digital Twins	Dassaul t System es	Siemen s			+
2.	Вартість обслуговуванн я					+	
3.	Відкритий код					+	
4.	Оновлення забезпечення						+

4.2 Технологічний аудит ідеї проекту

Таблиця 4.2 Технологічна здійсненність ідеї проекту

№	Ідея проекту	Технології та її реалізації	Наявність технології	Доступність технологій
1	Побудова фізичної симуляції	Digital Twins	+	+
2	Обмін даними	Websocket/R EST API	+	+
3	Відображення	WebGL	+	+

4.3 SWOT-аналіз проекту

Аналізуються можливості впровадження проекту на основі складання матриці SWOT-аналізу – матриці, де ідентифікуються сильні (Strength) та слабкі (Weak) сторони, загрози (Troubles) та можливості (Opportunities)

проекту (табл.). Перелік ринкових загроз та ринкових можливостей складається на основі аналізу факторів загроз та факторів можливостей зовнішнього середовища проекту. Перелік сильних та слабких сторін складається на основі аналізу внутрішніх факторів, притаманних організації (підприємству) та ініціатором проекту, а також береться до уваги досвід конкурентних компаній для отримання найбільш релевантних у існуючій ситуації даних.

Даний аналіз використовується для оцінки основних показників проекту і побудови стратегії на основі цих показників.

Таблиця 4.3.1 SWOT аналіз

<p>Сильні сторони:</p> <ul style="list-style-type: none"> - Вартість - Простота впровадження 	<p>Слабкі сторони:</p> <ul style="list-style-type: none"> - Відсутність функціоналу, який наявний у конкурентів - Відсутність інфраструктури Програмного забезпечення
<p>Можливості:</p> <ul style="list-style-type: none"> - Масштабованість - Використання у інших сферах - Інтеграція з існуючими системами - Утворення бібліотеки програмних рішень 	<p>Загрози:</p> <ul style="list-style-type: none"> - Кібератаки - Створення несумісних із продуктом технологій - Створення більш функціональних аналогів

Таблиця 4.3.2 Характеристика потенційних клієнтів стартап-проекту

№	Потреба	Цільова аудиторія	Вимогу споживачів до продукту
1	Отримати статистику про виробництво, або технологічні продукти у реальному часі, мати можливість оперувати даними	підприємства	Економічні та ергономічні
2		транспорт	
3		виробництва	

Таблиця 4.3.3 Фактори загроз

№	Фактор	Зміст загрози	Можлива реакція компанії
1	Втрата зв'язку із мережею Інтернет	Внаслідок порушення роботи мережі, дані з датчиків можуть не потрапити до єдиної бази даних на сервері	Налаштування локальної мережі із системою бекапування

Таблиця 4.3.4 Фактори можливостей

№	Фактор	Зміст можливості	Можлива реакція компанії
1	Формування незалежних об'єднань розробників	Створення незалежних додатків до програмного забезпечення	Отримання більшої кількості функціоналу
2	Зменшення штату	Автоматизація процесів	Економія коштів
3	Виявлення помилок у процесах	Удосконалення процесів	Економія за рахунок мінімізації ризиків та оптимізації процесів

Наступним кроком маркетингового аналізу є ступеневий аналіз конкуренції на ринку, основною задачею якого є визначення типу конкуренції, додаткових параметрів конкуренції, а також типу боротьби за клієнтів.

На основі даного кроку буде обиратися стратегія виходу на ринок, адже основна складність у створенні стартап-проекту – це отримання перших клієнтів за рахунок вдалої стратегії, у свою чергу, основою будь-якої стратегії виходу на ринок є саме стратегія конкуренції.

Таблиця 4.3.5 Ступеневий аналіз конкуренції на ринку

<i>Особливості конкурентного середовища</i>	<i>В чому проявляється дана характеристика</i>	<i>Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)</i>
Тип конкуренції	Олігополія	Диференціація у маркетингових підходах у взаємодії із споживачами
Рівень конкурентної боротьби	Національний рівень конкурентної боротьби	Відповідність технології та якості продукту вимогам законодавства України
За галузевою ознакою	Міжгалузева	Урахування особливостей конкурентного середовища у кожній галузі, з якою взаємодіє стартап.
Конкуренція за видами товарів	Товарно-видова конкуренція	Відстеження тенденцій на ринку з можливістю появи на ринку продуктів-замінників.
За характером конкурентних переваг	Цінова	Удосконалення технології, що спрямована на підвищення базових переваг.
За інтенсивністю	Не марочна інтенсивність	Забезпечення масштабованості стартапу в найближчій перспективі для створення стійкого сприйняття стартапу як окремої бізнес одиниці.

Як видно із таблиці 4.3.5, ступеневий аналіз конкуренції показав, що ринок має високий рівень конкуренції, що значить велику кількість потенційних покупців.

Більш того, такий вигляд конкурентної картини часто корелює із періодичним розширенням ринку та кількістю гравців на ньому, часто такий ринок називають “розігрітим ринком”.

Так як ми не маємо можливості здійснювати маркетингові маневри, та витратити значні кошти на рекламу та закупівлю маркетингових активностей, було обрано ціновий характер конкурентних переваг, адже ми маємо можливість ділити продукт на пакети, за рахунок його гнучкості та адаптивності, що дозволяє підібрати ідеальне рішення для кожного клієнта з точки зору його потреб та фінансових можливостей.

Таблиця 4.3.6 Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Навести перелік прямих конкурентів	Визначити бар'єри входження в ринок	Визначити фактори сили постачальників	Визначити фактори сили споживачів	Фактори загроз з боку замінників
Висновки:	Визначити інтенсивність конкурентної боротьби з боку прямих конкурентів	- чи є можливості входу в ринок? - чи є потенційні конкуренти? Строки виходу їх на ринок, а також строки отримання вагомого проценту від існуючого ринку.	Чи постачальники диктують умови роботи на ринку? Які?	Чи клієнти диктують умови роботи на ринку? Які?	Обмеження для роботи на ринку через товари замінники

Таблиця 4.3.7 Обґрунтування факторів конкурентоспроможності

№	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Економічність	Низька вартість процесу підтримки та впровадження
2	Гнучкість	Система може бути переобладнана через конструктор-подібну архітектуру
3	Адаптивність	Система може виконувати відразу декілька паралельних задач за рахунок можливості інтегрувати додати
4	Масштабованість	Систему можна розширювати за рахунок додавання мікро сервісів та модулів

Таблиця 4.3.8 Порівняльний аналіз сильних та слабких сторін

№	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з системою						
			-3	-2	-1	0	+1	+2	+3
1	Економічність	20							+
2	Гнучкість	18						+	
3	Адаптивність	16						+	
4	Масштабованість	19						+	

Як ми можемо бачити із таблиці 4.3.8, продукт має переваги в усіх факторах конкурентоспроможності, що означає, що дана цільова аудиторія більш всього підходить для стартап-проекту, а також, що даний проект покриває усі основні потреби обраної цільової аудиторії, це означає, що вартість отримання нового покупця буде значно меншою ніж у конкурентів в одному і тому ж сегменті ринку.

Таблиця 4.3.9 Альтернативи ринкового впровадження стартап-проекту

№	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Стратегія нейтралізації ринкових загроз сильними сторонами стартапу.	висока	6 місяців
2	Стратегія підсилення сильних сторін за рахунок ринкових можливостей.	середня	1 рік
3	Стратегія компенсації слабких сторін наявними ринковими можливостями.	середня	1 рік

4.4. Розроблення ринкової стратегії проекту

Розроблення ринкової стратегії передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів.

Головною цільовою групою стартап-проекту є середній бізнес, але через гнучкість і адаптивність проекту можна включити й інші групи, що мають подібні потреби та виробничо-технічну базу.

Важливою особливістю визначених груп є те, що всі вони представники бізнесу, що означає, що стартап-проект підпадає під категорію B2B (Business to Business). Ця категорія має особливості для бізнесу у маркетингових компаніях, типу ініціалізації продажу та методу здійснення угод, але це найбільш приваблива категорія із точки зору обраної сфери, адже, так званий “середній чек” – показник, що відображає усереднену суму однієї продажі, що платить клієнт – найвищий саме у обраній категорії B2B. Звісно, у великого бізнесу цей показник більший, але є вірогідність, що рекламні витрати не зможуть забезпечити достатній рівень маржинальності.

Таблиця 4.4.1 Альтернативи ринкового впровадження
стартап-проекту

№	Опис профілю цільової потенційних клієнтів групи	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи	Інтенсивність конкуренції в сегменті	Просота входу у сегмент
1	Великий бізнес	висока	специфічний	висока	високий бар'єр входу у галузь
2	Середній бізнес	висока	специфічний	середня	легкий вхід
3	Логістичні компанії	висока	специфічний	висока	високий бар'єр входу у галузь
4	Військові системи	середня	достатній	висока	високий бар'єр входу у галузь
5	Сортувальні системи для логістики	висока	достатній	висока	легкий вхід

Обрана стратегія: стратегія лідерства за витратами.

Стратегія охоплення ринку: ексклюзивна стратегія охоплення ринку.

Наступним кроком є визначення базової стратегії конкурентної поведінки, яка є основою стратегії виходу на ринок, адже саме конкурентна стратегія є основним параметром у отриманні клієнтів, а значить й в успішному виході на ринок. Базова стратегія визначається на основі методу пошуку клієнтів та кількості і типу існуючих на ринку гравців.

Таблиця 4.4.2 Визначення базової стратегії конкурентної поведінки

<i>Чи є проект першопрохідцем на ринку?</i>	<i>Чи буде компанія шукати нових споживачів або забирати існуючих у конкурентів?</i>	<i>Чи буде компанія копіювати основні характеристики товару конкурента, і які?</i>	<i>Стратегія конкурентної поведінки</i>
Так	50/50	Функціонал мережевого інтегрування	Оборонна

4.5. Обґрунтування ресурсів та витрат проекту

4.5.1 Визначення ціни

Таблиця 4.5.1 Визначення ціни

Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на продукцію	Розрахункова ціна продукції
\$50000	\$60000	\$100000 - \$5000000	30000 грн. - 100000 грн.	\$10000

Таблиця 4.5.2 Визначення обсягу продукції

Показник	Значення по роках				
	2021	2022	2023	2024	2025
Загальна потреба в продукції	2	5	8	30	70
Можливі річні обсяги випуску	0	0	8	30	70
Ціна одиниці продукції (тис. дол.)	0	0	10000	10000	10000
Річні обсяги випуску в	0	0	80000	300000	700000

4.5.2 Визначення загальних початкових інвестицій

Для визначення загальних інвестицій буде використовуватися розрахунок від необхідних витрат. Варто зазначити, що інвестиції вираховуються на етап до виходу на ринок, адже сфера у якій буде запущено стартап-проект постійно змінюється і з'являються нові гравці, що веде за собою можливість глобальної зміни маркетингових процесів

Таблиця 4.5.3 Розрахунок загальних початкових інвестиційних витрат

№	Назва етапу	Строк виконання	Обсяг фінансування, тис. грн
1	Проведення досліджень	3 місяці	120
2	Розробка прототипу датчиків	6 місяців	200
3	Розробка програмного забезпечення	6 місяців	200
4	Налагодження виробничого процесу датчиків	6 місяців	500
5	Розробка фреймворка для програмування датчиків	3 місяці	80
Разом:		24 місяці	1100

4.5.3 Розрахунок виробничих витрат

Було виділено декілька типів витрат, а саме:

- Витрати на аренду офіса
- Витрати на програмне забезпечення (Купівля програмного забезпечення для розробки, оплата ліцензії для фізичної симуляції, сплата ренти за сервера)
- Витрати на оплату праці (В своїй основі, будуть найматися програмісти та проектні менеджери)
- Витрати на маркетинг (Купівля реклами, оплата розробки корпоративного сайту, оплата роботи дизайнерів, друкування

- фізичної реклами, оплата разової роботи маркетингової рекламної агенції та SEO-агенції)
- Витрати на виробничий процес (Налагодження виробничого процесу датчиків, закупівля матеріалу для виробництва, страхування)

Таблиця 4.5.4 Розрахунок виробничих витрат

<i>Показник</i>	<i>Значення по роках, тис. грн.</i>				
	<i>2021</i>	<i>2022</i>	<i>2023</i>	<i>2024</i>	<i>2025</i>
Витрати на аренду офіса	15	50	50	50	50
Витрати на програмне забезпечення	10	10	100	100	100
Витрати на оплату праці	400	400	1200	1200	1200
Витрати на маркетинг	0	200	800	800	800
Витрати на виробничий процес	40	350	350	350	350
Разом	465	1210	2500	2500	2500

4.6 План робіт та партнери проекту

На основі проведеного маркетингового аналізу може бути проведено планування робіт перших етапів розвитку стартап-проекту. Для кожної роботи визначається виконавець, адже не завжди вигідно наймати штат працівників для виконання короткострокових або середньострокових задач, так як наймаючи працівників у штат ми йдемо на постійні витрати, що є небажаним для нового стартап-проекту.

Можна виділити чотири типи партнерських відносин: співробітництво між неконкуруючі компаніями; партнерство між конкурентами; спільні підприємства для запуску проектів; відносини виробника з постачальниками, підрядниками та виконавцями певних видів робіт по контракту.

Основною задачею є мінімізувати витрати за рахунок правильного делегування робіт, які не входять у експертну область стартап-проекту, або його засновників.

Таблиця 4.6 План робіт

№	Бізнес-процес	Термін	Виконавець	Результат
1	Проведення досліджень	3 місяці	Розробники	Обрання підходу до розробки
2	Розробка прототипу датчиків	6 місяців	ООО “Хладар”	Розробка прототипу для кожного із датчиків
3	Розробка програмного забезпечення	6 місяців	Розробники	Отримання програмної бази для продукту
4	Налагодження виробничого процесу датчиків	6 місяців	ООО “Хладар”	Отримання готового процесу виробництва
5	Розробка фреймворка для програмування датчиків	3 місяці	Розробники	Отримання технології для розробки ПО незалежними розробниками

4.7 Грошовий потік та економічна оцінка

У даному пункті будуть проведені розрахунки для визначення точки беззбитковості, що відображає обсяг продажів, при досягненні якого, компанія має прибуток більший ніж витратна частина, або, іншими словами, компанія має позитивний дохід.

Важливою частиною цього визначення є постійна тенденція.

Розрахунок точки беззбитковості можна виконати за допомогою наступної формули:

$$T = \frac{C}{P-V}.$$

Де C – постійні витрати, такі як оренда офісу або комунальні послуги. Постійні витрати на залежать від кількості вироблених товарів, або мають не прямо пропорційну залежність.

P – вартість вироблення одного продукту

V – змінні витрати, або динамічні витрати. Ці витрати змінюються прямо пропорційно із кількістю виробленої продукції, наприклад оплата праці працівників, які обслуговують верстати.

4.7.1 Формування грошового потоку

У даному розділі буде розраховано показник чистого дисконтованого доходу. Цей показник містить у собі різницю між прибутком проекту та залученими інвестиціями.

Таблиця 4.7 Формування грошового потоку

Показник	Значення по роках, тис. грн.					Разом
	2021	2022	2023	2024	2025	
Дохід	0	0	2200	8300	19500	30000
Витрати	465	1210	2500	2500	2500	9175
Прибуток	-465	-1210	-300	5800	17000	20825

Так як загальний розрахунковий прибуток більше нуля – проект можна вважати вигідним для інвестування.

4.7.2 Розрахунок індексу рентабельності інвестицій

ROI – це співвідношення інвестованих коштів до прибутку інвестора.

Вирахувати ROI можна за наступною формулою:

$$ROI = \sum_T D_t / \sum_T I_t.$$

Буде використовуватись саме ця формула, так як ми беремо до уваги періодичні інвестиції у декілька раундів.

D – надходження за період t

I – інвестиція за період t

Мета проекту – досягнути показника ROI більше ніж 2. Вважається, що нижчий рівень показника рентабельності інвестицій дорівнює одному, але при такому рівні інвестору не вигідно інвестувати у проект, так як існують альтернативні фонди для інвестування, що мають менші ризики та більші проценти (наприклад банківські депозити, або часне кредитування).

4.8 Висновки до розділу

Маркетинговий аналіз – ефективний метод аналізу проекту з точки зору бізнесу та прибутковості, який дозволяє надати переконливі обґрунтування щодо перспектив інвестування у проект.

Інвестування, у свою чергу, це один із найбільш ефективних методів отримання коштів для виходу проекту на ринок, адже інвестор однаково зацікавлений у тому, щоб проект успішно отримав перших споживачів. Таким чином, інвестор може бути залучений у процес розробки бізнес моделей, або маркетингового плану на одному рівні із засновником проекту у якості співзасновника.

Як і показано у розділі, маркетинговий аналіз дозволив визначити, що бізнес-модель проекту дозволяє повернути інвестиції у розмірі більш ніж вдвічі більшому розмірі, що вважається прийнятним результатом, та спонукає інвесторів прийняти участь у проекті.

ВИСНОВКИ

У ході виконання магістерської дисертації було розглянуто питання впровадження симуляції методом цифрового двійника на виробництво. Був проведений аналіз існуючих методів самої симуляції та методів впровадження симуляції цифрового двійника у виробництво, а також були розглянуті та проаналізовані альтернативні рішення для впровадження симуляції.

Перед реалізацією програмного забезпечення був проведений аналіз технологій та визначено найбільш відповідні до поставлених задач технології. Також була розроблена архітектура програмного забезпечення таким чином, щоб вона відповідала усім вимогам гнучкого, адаптивного і сучасного програмного забезпечення.

Більш того, були проаналізовані ризики у роботі продукту, можливі причини збоїв та прийняті відповідні рішення для підвищення відказостійкості програмного забезпечення, так як введення додаткових етапів бекапування.

На основі проведеного дослідження варто зазначити, що основна проблема поширення інтеграції виробництва із автономною системою статистики та контролю може бути вирішена за рахунок створення стандартизованих інфраструктури із програмного забезпечення та фізичних датчиків із однаковим інтерфейсом.

Даний підхід вже використовується у великому бізнесі, де корпорації створюють свої особисті стандарти для подібних систем, але його все ще важко використовувати для середнього та малого бізнесу через високу вартість впровадження рішення на кожне незалежне виробництво та важкість налаштування інтеграції системи у виробничий процес без його безпосередньої зміни.

Основна задача даної роботи – вирішити проблему доступності системи отримання статистики та управління виробництвом за рахунок

симуляції методом цифрового двійника для малого та середнього бізнесу була досягнута за рахунок рішень, які були описані вище.

Перелік посилань

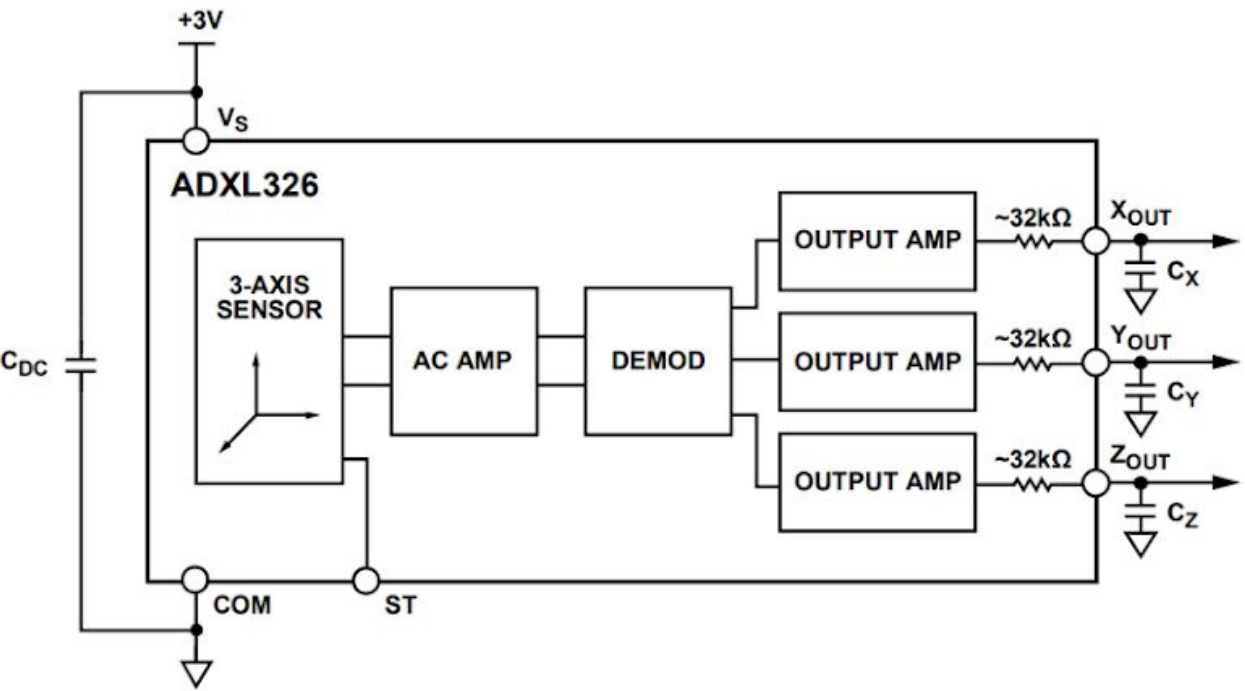
1. Документація Docker.
2. Документація Node.js.
3. Гончаров А. С., Саклаков В. М. Цифровий двійник: огляд існуючих рішень і перспективи розвитку технології. (2018).
4. Michael Grieves, Digital twin: Manufacturing excellence through virtual factory replication
5. S. N. Masaev, A. N. Minkin and D. A. Edimichev. An algorithm for determining the state of a non-stationary dynamic system for assessing fire safety control in an enterprise by the method of integrated indicators
6. Saddik, A. El (April 2018). "Digital Twins: The Convergence of Multimedia Technologies". IEEE MultiMedia.
7. Chhetri, Mohan Baruwal; Krishnaswamy, Shonali; Loke, Seng Wai (2004). Bussler, Christoph; Hong, Suk-ki; Jun, Woochun; Kaschek, Roland; Kinshuk; Krishnaswamy, Shonali; Loke, Seng Wai; Oberle, Daniel; Richards, Debbie (eds.). "Smart Virtual Counterparts for Learning Communities". Web Information Systems – WISE 2004 Workshops. Lecture Notes in Computer Science. Springer Berlin Heidelberg.
8. "Creating an Embedded Digital Twin: monitor, understand and predict Device Health Failure". Inn4mech - Mechatronics and Industry 4.0 Conference Presentation - 2018.
9. Indrawan, Maria; Ling, Sea; Smanchat, Sucha; Loke, Seng Wai (2008). "Formal Mirror Models: an Approach to Just-in-Time Reasoning for Device Ecologies".
10. Loke, S. W.; Thai, B. S.; Torabi, T.; Chan, K.; Deng, D.; Rahayu, W.; Stocker, A. (July 2015). "The La Trobe E-Sanctuary: Building a Cross-Reality Wildlife Sanctuary". 2015 International Conference on Intelligent Environments
11. "Azure Digital Twins". Microsoft.
12. "Digital twin to enable asset optimization". Smart Industry.

13. "What Are Digital Twins And Why Will They Be Integral To The Internet Of Things?". ARC. Retrieved 26 July 2017.
14. Grieves, M. and J. Vickers, Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems, in Trans-Disciplinary Perspectives on System Complexity, F.-J. Kahlen, S. Flumerfelt, and A. Alves, Editors.
15. Glaessgen, Edward, and David Stargel. "The digital twin paradigm for future NASA and US Air Force vehicles." 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA. 2012.
16. Tao, Fei, et al. "Digital twin-driven product design framework." International Journal of Production Research.
17. Thilmany, Jean (September 21, 2017). "Identical Twins". ASME. Retrieved September 10, 2019.
18. "Digital twins – rise of the digital twin in Industrial IoT and Industry 4.0". i-SCOOP.
19. Greengard, Samuel. "Digital Twins Grow Up". cacm.acm.org. Retrieved 2019-09-11.

ДОДАТКИ

ДОДАТОК А

IK-91.17 3132.002 B3

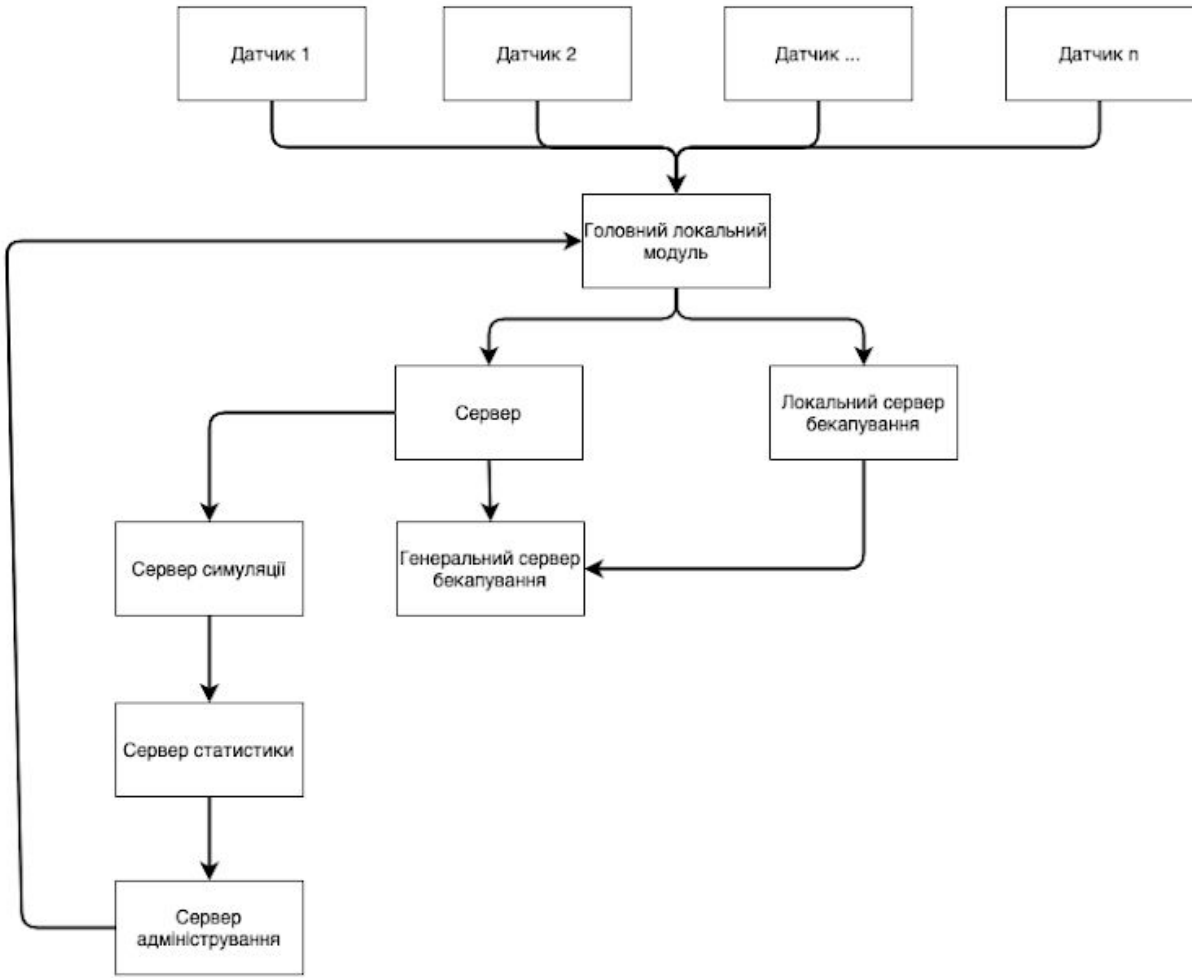


Підпис і дата	
Ім'я, № дубл.	
Взам. ім'я, №	
Підпис і дата	
Ім'я, № ориг.	

					ІК-91.17. 3132.002 ВЗ				
					Діаграма прецедентів	Літ.	Маса	Мірило	
Зм.	Лист	№ докум	Підпис	Дата					
Розроб.		Сізов Д.С.							
Перев.		Поліщук М.М.							
					Кафедра Технічної кібернетики	Лист	1	Листів	1
Н.контр		Пасько В.П.				Група ІК-91мп			
Затв.		Пархомей І.Р.							

ДОДАТОК Б

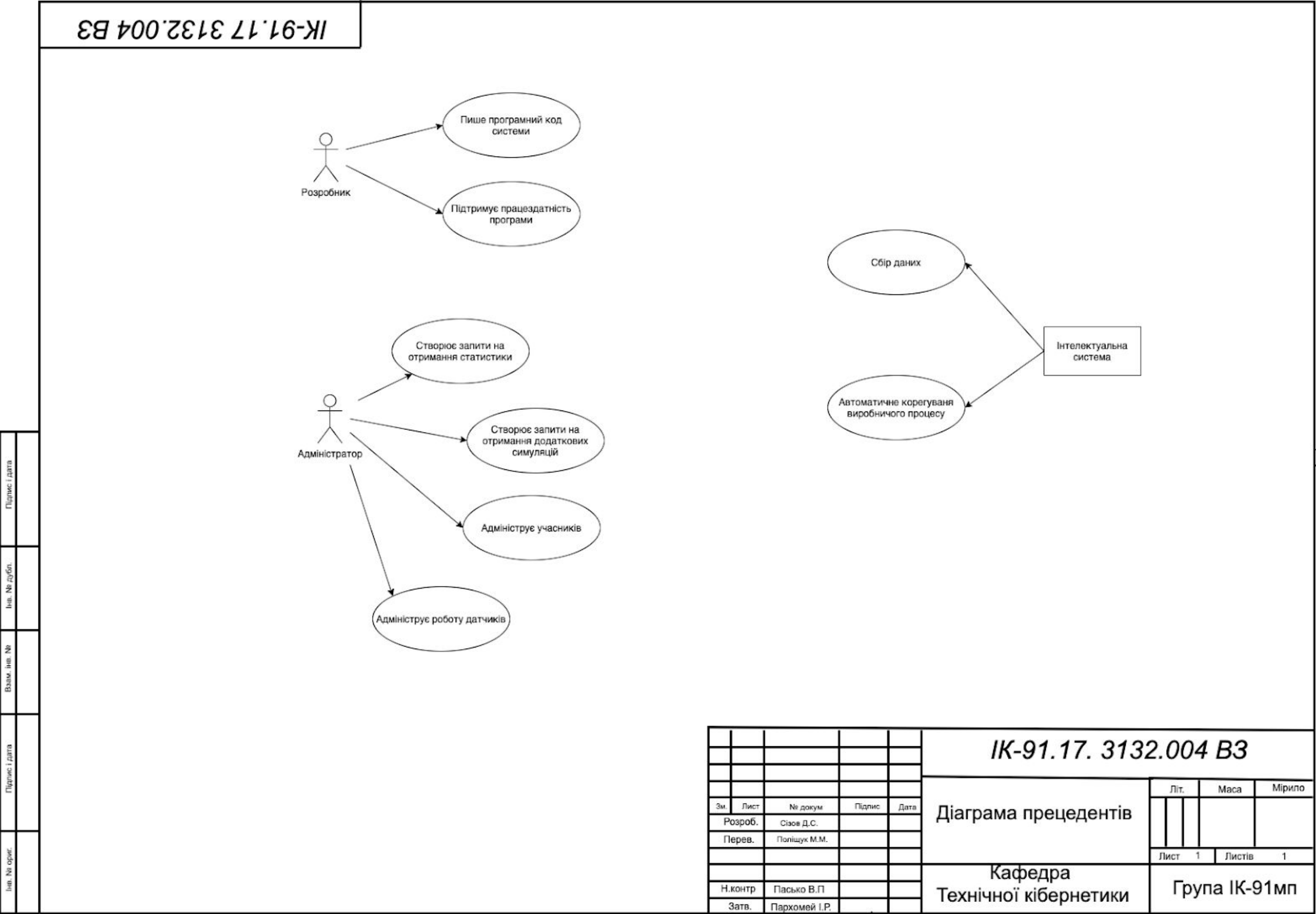
IK-91.17 3132.003 B3



Ім'я, № ориг.	
Підпис, дата	
Взам. ім'я, №	
Ім'я, № дубл.	
Підпис, дата	

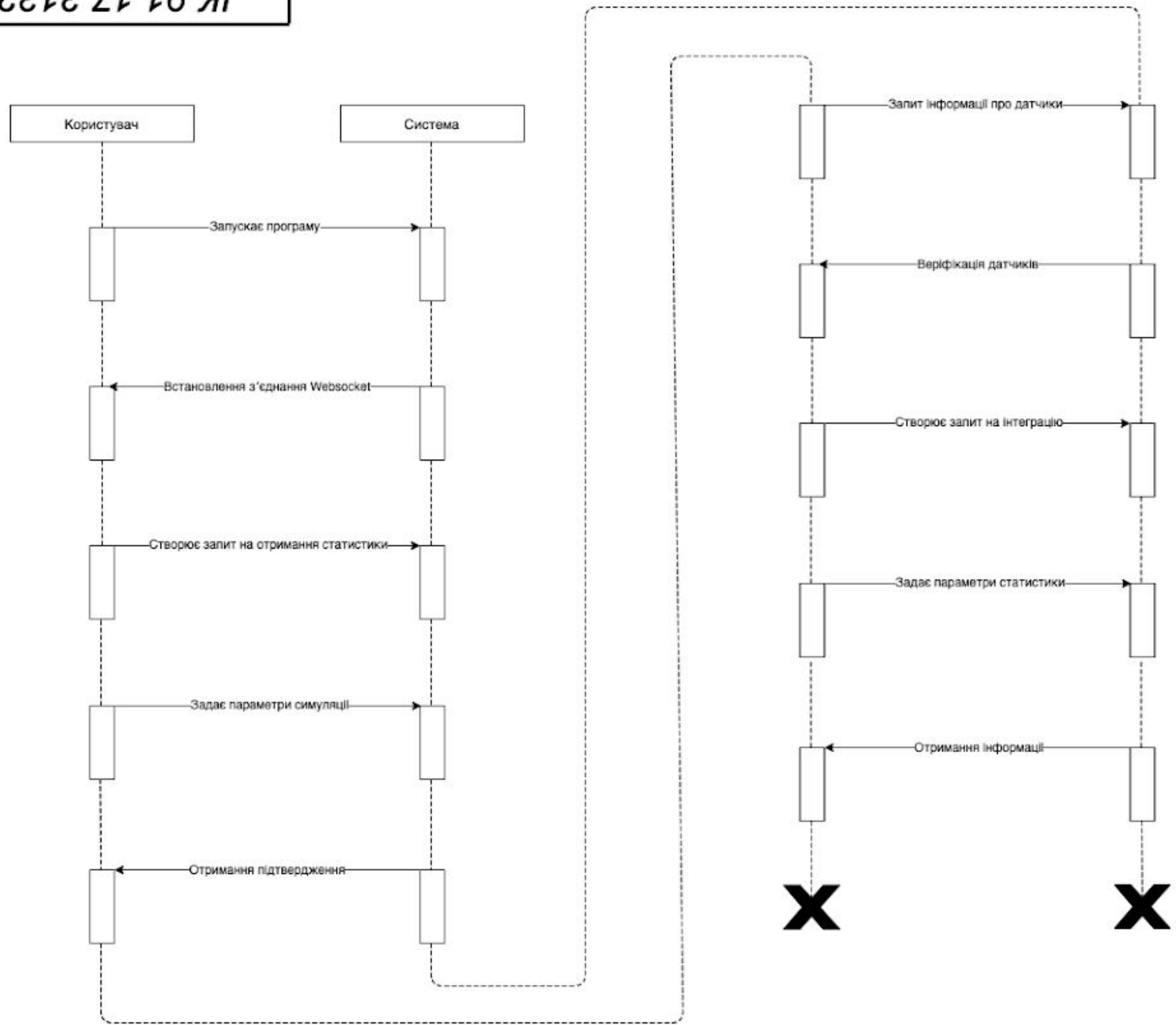
					ІК-91.17. 3132.003 ВЗ				
					Діаграма прецедентів	Літ.		Маса	Мірило
Зм.	Лист	№ докум	Підпис	Дата					
Розроб.		Сізов Д.С.							
Перев.		Поліщук М.М.							
						Лист 1		Листів 1	
Н.контр		Пасько В.П			Кафедра Технічної кібернетики	Група ІК-91мп			
Затв.		Пархомей І.Р.							

ДОДАТОК В



ДОДАТОК Г

IK-91.17 3132.004 B3



X

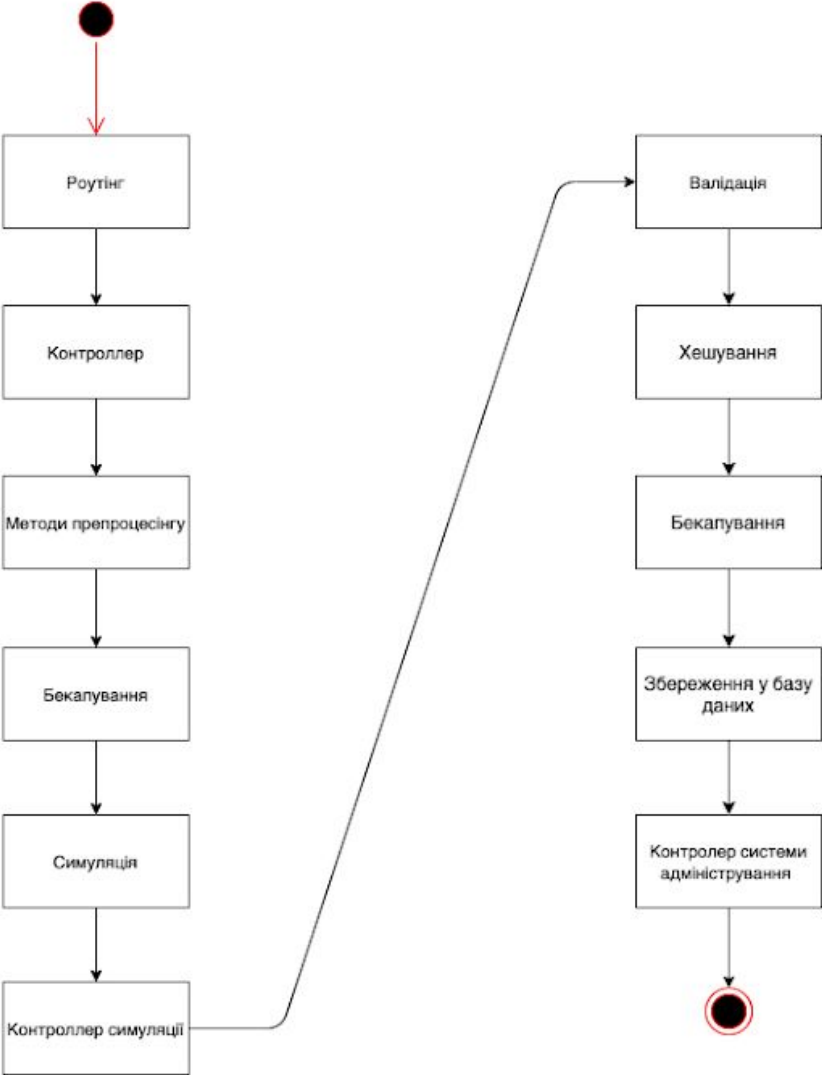
X

Підпис / дата	
Інв. № дубл.	
Взам. інв. №	
Підпис / дата	
Інв. № ориг.	

					IK-91.17. 3132.004 B3				
					Діаграма прецедентів	Літ.		Маса	Мірило
Зм.	Лист	№ докум	Підпис	Дата	Кафедра Технічної кібернетики	Лист 1		Листів 1	
Розроб.	Сізов Д.С.								
Перев.	Поліщук М.М.								
Н.контр	Пасько В.П				Група ІК-91мп				
Затв.	Пархомай І.Р.								

ДОДАТОК Д

IK-91.17 3132.006 B3



Підпис і дата	
Ім'я, № дубл.	
Взам. ім. №	
Підпис і дата	
Ім'я, № ориг.	

					IK-91.17. 3132.006 B3												
					Діаграма прецедентів					Літ.		Маса		Мірило			
Зм.	Лист	№ докум.	Підпис	Дата	Кафедра Технічної кібернетики					Лист 1		Листів 1		Група ІК-91мп			
Розроб.		Слов Д.С.															
Перев.		Попіщук М.М.															

ДОДАТОК Е

[illegible]